Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2017

Performance Assessment of Massive MIMO Systems for Positioning and Tracking of Vehicles in Open Highways

Markus Petersson

LINKÖPING UNIVERSITY

Master of Science Thesis in Electrical Engineering

Performance Assessment of Massive MIMO Systems for Positioning and Tracking of Vehicles in Open Highways

Markus Petersson

LiTH-ISY-EX--17/5049--SE

Supervisors: **Lic. Christopher Mollén**
ISY, Linköpings universitet
Dr. Mário Costa
Huawei Technologies

Examiner: **Prof. Erik G. Larsson**
ISY, Linköpings universitet

Division of Communication Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Till min farfar Erik,
min morfar Bertil
och min gudfar Nils

Abstract

The next generation of mobile networks (5G) is currently being standardized, and massive MIMO (Multiple-Input-Multiple-Output) is a strong candidate to be part of this standard. Other than providing higher data rates and lower latency, high accuracy positioning is also required. In this thesis, we evaluate the achievable performance of positioning using massive MIMO systems in open highway scenarios. Relevant theory from sensor array signal processing and Bayesian filtering is presented, and is used in a simulation environment on large antenna arrays representing massive MIMO base stations. Positioning is done by utilizing the uplink pilot reference signals, where the Direction of Arrival (DOA) of the pilot signal is estimated, and then used for position estimation. Estimation of the DOA is done by both a maximum-likelihood method and by using an Extended Kalman Filter (EKF). A positioning error of less than 8 m is achieved with absolute certainty when the vehicle is less than 300 m from the base station. It is also concluded that this result could be improved by using more sophisticated filtering algorithms.

Acknowledgments

This thesis concludes my five years at Linköping University, and what a ride it has been. The completion of this thesis calls for acknowledgement of some of the people that has made it possible.

First and foremost I would like to thank Dr. Mário Costa, my supervisor from Huawei Technologies, for giving me the opportunity to do this thesis work, for always having answers to my questions, and for always being positive and supportive. I would also like to express my sincere gratitude to Lic. Christopher Mollén, my supervisor at the Division of Communication Systems, for always having an open door, encouraging discussion, and for his precise and honest comments on my work. I am also indebted to my examiner Prof. Erik G. Larsson, whose encouragement, involvement, and sharp eye has been both inspirational and helpful. I am especially grateful for the discussions we had regarding nuisance parameters and the Cramér-Rao Bound. I would like to thank Ema Becirovic, with whom I have shared an office, for the countless coffee-breaks, for her help with the small things regarding the thesis, and for at times taking my mind off it. Furthermore, I would also like to thank everyone in the Division of Communication Systems for all being nice and helpful.

I would also like to take the opportunity of thanking all the people I have met and all the friends I have made during my five years in Linköping. It is you who have made these some of the best, and some of the most memorable years of my life.

Finally, I would like to thank my immediate family for always believing in me and for always supporting me. For this I will always be grateful. You have had a bigger part than you think in the making of this thesis.

Linköping, June 2017
Markus Petersson

Contents

Notation

SETS

Notation	Meaning
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\mathbb{C}^{n\times m}$	The set of complex $n \times m$ matrices
$\boldsymbol{\Omega}$	The set of parameters that parameterize the steering vector
$\boldsymbol{\xi}$	The set of nuisance parameters
$\boldsymbol{\Psi}$	The union of $\boldsymbol{\Omega}$ and $\boldsymbol{\xi}$

MATHEMATICAL NOTATION

Notation	Meaning
$(\,\cdot\,)^{\dagger}$	The Moore-Penrose Pseudo Inverse
$\mathcal{H}$	A Hilbert space
$(\,\cdot\,)^{H}$	The Hermitian transpose
$[\boldsymbol{v}]_n$	The n-th element of a vector $\boldsymbol{v}$
$[\boldsymbol{M}]_{n,m}$	The (n,m)-th element of a matrix $\boldsymbol{M}$
$(\,\cdot\,)^{T}$	Transpose
$\Re\{\,\cdot\,\}$	Real part
$\otimes$	Kronecker Product
$\odot$	Schur Product
$(\,\cdot\,)^{*}$	Complex Conjugate
$\mathrm{tr}(\,\cdot\,)$	Trace Operator
∇_{ω}	Gradient with respect to the variables in $\boldsymbol{\Omega}$
∇_{ψ}	Gradient with respect to the variables in $\boldsymbol{\Psi}$

ABBREVIATIONS

Abbreviation	Meaning
DOA	Direction of Arrival
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
DML	Deterministic Maximum Likelihood
SML	Stochastic Maximum Likelihood
KF	Kalman Filter
EKF	Extended Kalman Filter
TTI	Transmission Time Interval
UE	User Equipment
FIM	Fisher Information Matrix
CRB	Cramér-Rao Bound
MST	Manifold Separation Technique
PDF	Probability Density Function
MSE	Mean Square Error
RMSE	Root Mean Square Error
CCDF	Complementary Cumulative Distribution Function
SNR	Signal-to-Noise-Ratio
LTE	(4G) Long Term Evolution

1 Introduction

This thesis is about how a massive MIMO base station can be used to estimate and track the position of a single user in an open highway. The concept of massive MIMO is briefly described in this chapter, where the specific questions we answer are also given. Furthermore, we compare the contributions of this thesis to earlier work.

1.1 Future Cellular Communication

In the last five years, the amount of mobile data traffic has grown eighteen-fold and, in 2016 alone, Global mobile data traffic grew by 63%. Hundreds of millions of new mobile devices and connections are yearly added, and the amount of global mobile data traffic is projected to increase from 7.2 exabytes per month in 2016 to 49 exabytes per month in 2021 [5].

The new generation of mobile networks (5G) is currently being standardized in order to meet the quite remarkable goal in increased mobile data-traffic. Massive MIMO is one of the technologies competing to be the technical enabler of 5G. The concept of massive MIMO was proposed in [15], where the author considers a base station with an infinite number of antennas. Later, research has shown that massive MIMO requires neither an infinite number of antennas, nor expensive high-end hardware to achieve its benefits [2]. This makes massive MIMO viable in practice. Researchers at Lund University and the University of Bristol have together constructed a real-time testbed for massive MIMO, which has displayed both great throughput and low latency [22], making another case for massive MIMO as a candidate for 5G.

In massive MIMO, a base station is equipped with a large number of antennas, and serves several dozens of users simultaneously. In order to make massive MIMO scalable, the users transmit pilot reference signals (signals with a known structure), for channel estima-

tion. The pilot signals are in this thesis assumed to be narrowband signals. Narrowband signals impinging on an antenna array is a well-studied problem in sensor array signal processing, where explicit estimators for determining the incident angles [12] of impinging narrowband signals are known. Since the incident angles of the pilot signals can be estimated, positioning using only a single massive MIMO base station is also possible, if we assume a height of the user equipment (UE) and have a line-of-sight-scenario.

Although the main reason for the development of 5G is handling the increase in mobile data traffic, there are also other requirements that the upcoming standard is to fulfill. One such requirement is high accuracy positioning [17], which is partly motivated by the large industrial interest in autonomous vehicles (driver-less cars).

In the last few years, there has been an enormous industrial investment into autonomous vehicles. In Gothenburg, Volvo is already doing trials where fully autonomous vehicles are driving on public roads [4], and companies such as Zenuity have emerged, who solely work with technology involving autonomous vehicles [11].

There is still a lot of work and trials to be done before autonomous vehicles will be commercially available and entitled to driving everywhere. One important technological feature that would help in hastening this development is more accurate positioning, both from a security and a traffic efficiency perspective.

Consider the following scenario: Alice and Bob are both riding in autonomous vehicles in opposite directions on an open highway. In front of Alice's vehicle there is a slow driving vehicle that her vehicle needs to overtake. Alice's and Bob's vehicles are aware of the potential crash, since they have exchanged information about both position and velocity. The vehicles can, based on this information, schedule course and speed accordingly, enabling Alice's vehicle to overtake the hindrance safely. It is self evident that the more accurate the exchanged information is, the more secure the overtaking would be.

The development and deployment of systems capable of positioning might be considered an infringement of personal integrity. However, such systems are already in place, which is why this thesis work would not enable any additional infringement. Instead, the possible improved accuracy in positioning could, e.g. be used in saving lives by improving the security of future autonomous vehicles.

The aim of this thesis is to conduct a performance assessment of massive MIMO systems for positioning in open-highways, in order to investigate whether it can live up to the positioning requirements of the next generation mobile networks is expected to provide.

1.2 Problem Statement

We will answer the questions:

- What is the positioning accuracy a massive MIMO system can achieve in open highways?
- Does increasing the number of base station antennas improve positioning accuracy?
- How should base stations be deployed to maintain a high positioning accuracy?

1.3 Limitations

This work focuses on a scenario where:

- there was only one one vehicle on the highway. Thus, no pilot contamination had to be handled.
- the base station antennas were modeled as dipoles. Polarization was not taken into account.
- positioning was only done for a line-of-sight scenario. In real life, there might not always be a direct path to a base station.
- we only studied base stations with 25 antennas and 100 antennas.
- the results in this thesis are bound to the algorithm that we have used, where we assume planar waves.
- the base station antenna array is lying on the XZ-axis, thus there is an ambiguity in the azimuth angle. This ambiguity was assumed to be known.

1.4 Novel Contributions

The contributions in this thesis consists in applying the theory of sensor signal processing and Kalman filtering to massive MIMO antenna arrays. From this we have drawn conclusions regarding:

- positioning performance of a single massive MIMO base station
- the performance improvement obtained by increasing the number of antennas
- the future deployment of base stations, with regard to required positioning accuracy

1.5 Thesis Overview

This thesis is divided into the following chapters:

Chapter 1, Introduction: In this chapter, we give an introduction to the topic of the thesis, along with some motivation. The research questions to be answered are formulated in this chapter.

Chapter 2, Theory: In this chapter, we present the theory used in this thesis. We introduce a system model, derive estimators for the DOAs, present appropriate optimization algorithms for DOA estimates, and finally give some background in Bayesian filtering.

Chapter 3, Simulation Setup: In this chapter, we present the simulation setup. We give the models used for the base station, the UE and for the channel.

Chapter 4, Simulations: In this chapter, we provide extensive simulations, both to verify that the derived estimator bounds hold and are attained, and to evaluate the performance

of positioning and tracking of Massive MIMO systems using the methods presented in chapter 2. We discuss and compare the results for different simulation parameters.

Chapter 5, Conclusion: In this chapter we conclude the thesis by giving answers to the questions posed in the first chapter. We also give some suggestions for future research, continuing where this thesis was concluded.

2
Theory

In this chapter we introduce the theoretical concepts used throughout this thesis. The different parts are structured to be read sequentially, since each section builds on the theory from previous ones. The articles cited in this thesis has mainly been either original articles or peer-reviewed articles in sensor array signal processing, which is why they are considered both appropriate to use and credible. Textbooks are cited to provide references to theory that can be considered generally known.

2.1 Sensor Array Signal Processing

In sensor array signal processing, the problem of extracting information from the output of a collection of spatially distributed sensors is studied [21]. In this thesis, we study the case where only one signal is impinging on the sensor array, although, in general, it is assumed that the number of signals impinging on the array is finite. The signal is assumed to be a function of a finite number of parameters. Our goal is to estimate these parameters, which in our case will be the Direction of Arrival (DOA) of the impinging signal, and use this estimate in order to track vehicles driving in open highways.

2.1.1 Modeling the Sensor Array Output

In this section, we present a model of the sensor array output and the underlying assumptions. We show that the problem of parameterizing the impinging signal is in essence equivalent to parameterizing the time at which the signal arrives to each sensor, also called the time of arrival (TOA), relative to some reference point.

Consider a single narrowband signal, $\tilde{s}(t)$, transmitted on carrier frequency f_c, impinging on a sensor array comprising L sensors, as depicted in Figure 2.1. In general $\tilde{s}(t)$ is considered to be unknown and deterministic.

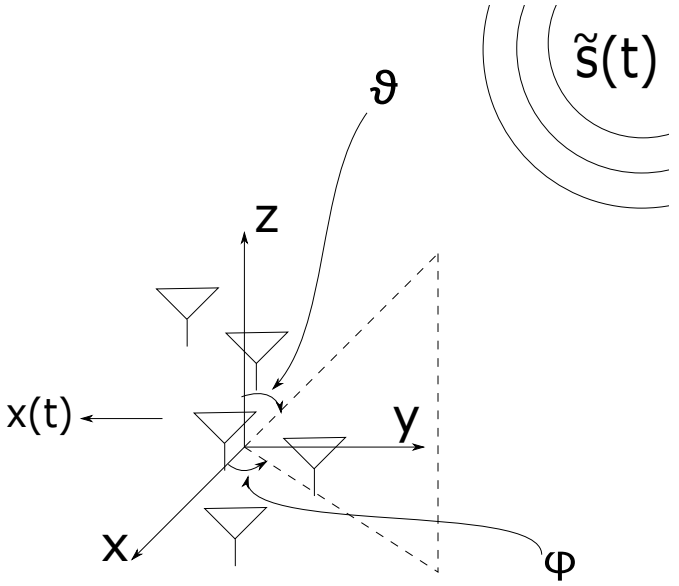

Figure 2.1: *Illustration of the setup where a single signal $\tilde{s}(t)$ impinges on a sensor array with* DOA (ϑ, φ)*, producing the output $x(t)$.*

In complex baseband, the noise-free output, $x_i(t)$, at sensor i, is modeled as

$$x_i(t) = \alpha \tilde{s}(t - \tau_i) \triangleq s(t - \tau_i), \tag{2.1}$$

where α is a complex scaling factor, and $\tau_i \in \mathbb{R}$ is the delay at antenna i relative to some reference point. Note that the complex scaling factor α is assumed to be the same for all antennas. This is an approximation that is valid when the distance to the source is much larger than the spatial spread of the antennas in the array. A change in reference point would only correspond to a phase shift, which can be included in α. Since the impinging signal is assumed to be narrowband, equation (2.1) can then be approximated as

$$x_i(t) = s(t - \tau_i) \approx e^{-j2\pi\tau_i f_c} s(t). \tag{2.2}$$

The sensor output was assumed to be parametrized by a finite number of parameters. Denote the set of these parameters $\mathbf{\Omega}$. By analyzing (2.2), we see that the only thing not independent of $\mathbf{\Omega}$ is the relative time delay, τ_i. Thus, we conclude that parameterizing the sensor output is equivalent to parameterizing the relative time delay. Thus we write (2.2) as

$$x_i(t) = \underbrace{e^{-j2\pi\tau_i(\mathbf{\Omega})f_c}}_{a_i(\mathbf{\Omega})} s(t) = a_i(\mathbf{\Omega}) s(t). \tag{2.3}$$

We can then write the output for the array as

$$\boldsymbol{x}(t) = \boldsymbol{a}(\mathbf{\Omega}) s(t), \tag{2.4}$$

where $\boldsymbol{a}(\mathbf{\Omega}) = \begin{bmatrix} a_1(\mathbf{\Omega}) & \dots & a_L(\mathbf{\Omega}) \end{bmatrix}^T$. In order to account for noise in our model, we

introduce additive noise. The noise, $\boldsymbol{n}(t)$, is assumed to be circularly-symmetric complex white Gaussian, i.e.

$$\boldsymbol{n}(t) \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}).$$

Under these assumptions, we expand the model in (2.4) and obtain

$$\boldsymbol{x}(t) = \boldsymbol{a}(\boldsymbol{\Omega})s(t) + \boldsymbol{n}(t), \tag{2.5}$$

which is our final model of the array output.

2.1.2 Parametrization of the Steering Vector by DOAs Under the Geometric Far-Field Assumption

If the UE emitting $s(t)$ is placed far away from the array, relative to the size of the array, the incoming wave can be approximated as a planar wave. If so is the case, then the DOA of the planar wave is identical for all sensors in the array. It is therefore natural to parameterize the relative time delay, τ_i, by the DOA, i.e. $\boldsymbol{\Omega} = \{\vartheta, \varphi\}$, where ϑ and φ are the co-elevation and azimuth directions of arrival respectively.

By geometric reasoning, and by neglecting a possible clock-offset between antenna i and the UE, we get the time delay, using the UE as reference point, can be written as

$$\tau_i^{\text{UE}} = \frac{1}{c}\left(\sin(\vartheta)\cos(\varphi)(x - x_i) + \sin(\vartheta)\sin(\varphi)(y - y_i) + \cos(\vartheta)(z - z_i)\right), \tag{2.6}$$

where (x, y, z) and (x_i, y_i, z_i) are the coordinates of UE and antenna i respectively, and c is the speed of light in vacuum.

In this work we do not assume that the base station is synchronized with the UE. However, we do assume that all the sensors within the array are synchronized. I.e. a clock-offset, ε, is introduced to the expression in (2.7), and we get

$$\tau_i^{\text{UE}} = \frac{1}{c}\left(\sin(\vartheta)\cos(\varphi)(x - x_i) + \sin(\vartheta)\sin(\varphi)(y - y_i) + \cos(\vartheta)(z - z_i)\right) + \varepsilon. \tag{2.7}$$

However, since the clock-offset ε relative the UE is equal for all antennas, we can cancel the effect of the clock-offset by choosing one of the antennas as reference for the time delay. It does not matter which antenna is chosen. The time delay at antenna i relative to e.g. antenna 1 is then simply

$$\tau_i = \tau_i^{\text{UE}} - \tau_1^{\text{UE}} = \frac{1}{c}\left(\sin(\vartheta)\cos(\varphi)(x_1 - x_i) + \sin(\vartheta)\sin(\varphi)(y_1 - y_i) + \cos(\vartheta)(z_1 - z_i)\right) \tag{2.8}$$

2.1.3 Wavefield Modeling

In order to make it easier to find the derivatives of the steering vector, which will later be needed in optimization and tracking, we now introduce the wavefield modeling formalism. Another advantage with using this formalism is that we do not have to make new derivations of the derivatives if we would change the array. In the wavefield modeling formalism, the output of the antenna array is decomposed into two factors. The first factor is the wavefield independent matrix called the *Sampling matrix*. The term "wavefield independent" means that no information regarding the impinging wavefield can be found

in the sampling matrix, and that it only depends on the array geometry as well as the beam patterns of the antenna elements. The second factor is the array independent vector called the *Coefficient vector* [8]. The term "array independent" means that no information regarding the properties of the array can be found in the coefficient vector, and that it only depends on the impinging wavefield. This means that we can write the output of the array as [8]

$$\boldsymbol{x}(t) = \underbrace{\boldsymbol{\mathcal{G}}\boldsymbol{d}(\boldsymbol{\Omega})}_{\boldsymbol{a}(\boldsymbol{\Omega})} s(t) + \boldsymbol{n}(t), \tag{2.9}$$

where $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{d}(\boldsymbol{\Omega})$ denotes the sampling matrix and coefficient vector respectively. This factorization of the array steering vector is sometimes called *manifold separation technique* (MST) [1]. In this thesis, the MST is used in the simulator to model the array steering vector.

The factorization used in the MST is simply a decomposition, of each element of the steering vector, into some set of basis functions, i.e.

$$\boldsymbol{a}_i(\boldsymbol{\Omega}) = \sum_{n=1}^{\infty} g_{i,n}\phi_n, \tag{2.10}$$

where $\boldsymbol{a}_i(\boldsymbol{\Omega})$ is the i-th element of the steering vector, ϕ_l, $l = 1, \ldots,$ are the basis functions, and $g_{i,l}$, $l = 1, \ldots,$ are the coefficients. Next, we will give an example of such a decomposition. In the example we give, the MST is done using a Fourier basis. This example can also be found in [1], but is included here since it is very illustrative.

Example 2.1

Consider azimuth DOA estimation, i.e. $\boldsymbol{\Omega} = \{\varphi\}$, for an arbitrary sensor array. Here, the co-elevation is assumed to be known, and $\vartheta = \frac{\pi}{2}$. Fix a cartesian coordinate system in the centroid of the array. The i-the element of the steering vector is given by

$$\boldsymbol{a}_i(\boldsymbol{\Omega}) = e^{-j2\pi f_c \tau_i(\varphi)} = e^{j\kappa r_i \cos(\varphi - \gamma_i)}, \tag{2.11}$$

where r_i is the distance from the centroid of the array to the i-th sensor, and γ_i is the angular coordinate of the same sensor, counted counter-clockwise from the x-axis. Here, the relative time-delay is given using the centroid of the array as the reference point, and $\kappa = \frac{2\pi f_c}{c}$ is the angular wave number. Equation (2.11) can be rewritten as [1]

$$\begin{aligned} e^{j\kappa r_i \cos(\varphi-\gamma_i)} &= \sum_{m=-\infty}^{\infty} j^m J_m(\kappa r_i) e^{jm(\gamma_i - \varphi_i)} \\ &= \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} \underbrace{\sqrt{2\pi} j^m J_m(\kappa r_i) e^{jm\gamma_i}}_{\triangleq [\mathcal{G}(r_i,\gamma_i)]_{i,m}} e^{-jm\varphi} \\ &= \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} \left[\mathcal{G}(r_i,\gamma_i)\right]_{i,m} e^{-jm\varphi}, \end{aligned} \tag{2.12}$$

where $\mathcal{G}(r_i, \gamma_i)$ is the (i, m)-th element of the sampling matrix and J_m is the Bessel function of the first kind of order m. From (2.12) we see that the sampling matrix is independent of the impinging wavefield, and depends only on the geometry on the array.

In order to obtain the standard form of the MST, as given in (2.9), we write the result from (2.12) on matrix form as

$$\boldsymbol{a}(\varphi) = \boldsymbol{\mathcal{G}}\boldsymbol{d}(\varphi), \tag{2.13}$$

where the (i, m)-th element of the sampling matrix is given by

$$[\mathcal{G}(r_i, \gamma_i)]_{i,m} = \sqrt{2\pi} j^m J_m(\kappa r_i) e^{jm\gamma_i}, \tag{2.14}$$

and the m-th element of $\boldsymbol{d}(\varphi)$ is given by

$$[\boldsymbol{d}(\varphi)]_m = \frac{1}{\sqrt{2\pi}} e^{-jm\varphi}, \quad m = \ldots, -1, 0, 1, \ldots. \tag{2.15}$$

In MST, the sampling matrix is viewed as an operator $\boldsymbol{\mathcal{G}} : \mathcal{H} \to \mathbb{C}^{L\times 1}$, while $\boldsymbol{d}(\boldsymbol{\Omega}) \in \mathcal{H}$ contains the basis functions [6]. The sampling matrix provides us with a full description of the antenna array, including geometry and imperfections e.g. from the manufacturing process, and the coefficient vector provides us with a full description of the wavefield. Note that all information about the received wavefield is characterized by the coefficient vector [7].

Note that both $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{d}(\boldsymbol{\Omega})$ must be of infinite dimension if (2.9) is to hold, since it requires an infinite number of basis functions, and therefore coefficients, to represent an element of the steering vector. However, in practice the steering vector can be modeled using a finite number of basis functions without sacrificing much accuracy. Moreover, by increasing the number of basis functions used, the truncation error can be made arbitrarily small. In [8] it was shown that the truncation error decays superexponentially as the number of basis functions tend to infinity.

2.1.4 Array Manifold Separation Using a 2-D Fourier Basis

In section 2.1.3, we saw how the wavefield modeling formalism allowed us, by writing the steering vector in a basis of orthonormal functions, to write the steering vector as a product between the sampling matrix $\boldsymbol{\mathcal{G}}$ and the coefficient vector $\boldsymbol{d}(\boldsymbol{\Omega})$. However, we did not specify the basis used in this decomposition.

One possible choice of basis, when $\boldsymbol{\Omega} = \{\vartheta, \varphi\}$, is the 2D Fourier functions. Define

$$[\boldsymbol{d}(\vartheta)]_{m_e} = \frac{1}{\sqrt{2\pi}} e^{jm_e\vartheta}, \quad m_e = \ldots, -1, 0, 1, \ldots, \tag{2.16}$$

$$[\boldsymbol{d}(\varphi)]_{m_a} = \frac{1}{\sqrt{2\pi}} e^{jm_a\varphi}, \quad m_a = \ldots, -1, 0, 1, \ldots, \tag{2.17}$$

and

$$\boldsymbol{d}(\vartheta, \varphi) = \boldsymbol{d}(\varphi) \otimes \boldsymbol{d}(\vartheta). \tag{2.18}$$

Here, m_e and m_a are mode indices in co-elevation and azimuth dimensions respectively, and $\otimes$ denotes the Kronecker product. The steering vector can then, in this 2D Fourier

basis, be decomposed as [6]

$$\boldsymbol{a}(\vartheta,\varphi) = \boldsymbol{\mathcal{G}}\boldsymbol{d}(\vartheta,\varphi), \tag{2.19}$$

where again $\boldsymbol{\mathcal{G}}$ is the sampling matrix.

An advantage gained by using the MST with a Fourier basis decomposition of the steering vectors is that the derivatives of the steering vector, which can be used in the computation of the Cramér-Rao bound (introduced in section 2.2.3), are straightforward to find. The derivatives of $\boldsymbol{a}(\vartheta,\varphi)$ with respect to ϑ and φ are

$$\begin{aligned}
\frac{\partial}{\partial\vartheta}\left(\boldsymbol{a}(\vartheta,\varphi)\right) &= \frac{\partial}{\partial\vartheta}\left(\boldsymbol{\mathcal{G}}\left[\boldsymbol{d}(\varphi)\otimes\boldsymbol{d}(\vartheta)\right]\right) \\
&= \boldsymbol{\mathcal{G}}\left[\frac{\partial}{\partial\vartheta}\left(\boldsymbol{d}(\varphi)\otimes\boldsymbol{d}(\vartheta)\right)\right] \\
&= \boldsymbol{\mathcal{G}}\left[\frac{\partial}{\partial\vartheta}\left(\boldsymbol{d}(\varphi)\right)\otimes\boldsymbol{d}(\vartheta) + \boldsymbol{d}(\varphi)\otimes\frac{\partial}{\partial\vartheta}\left(\boldsymbol{d}(\vartheta)\right)\right] \\
&= \boldsymbol{\mathcal{G}}\left[\boldsymbol{d}(\varphi)\otimes\frac{\partial}{\partial\vartheta}\left(\boldsymbol{d}(\vartheta)\right)\right],
\end{aligned} \tag{2.20}$$

and

$$\begin{aligned}
\frac{\partial}{\partial\varphi}\left(\boldsymbol{a}(\vartheta,\varphi)\right) &= \frac{\partial}{\partial\varphi}\left(\boldsymbol{\mathcal{G}}\left[\boldsymbol{d}(\varphi)\otimes\boldsymbol{d}(\vartheta)\right]\right) \\
&= \boldsymbol{\mathcal{G}}\left[\frac{\partial}{\partial\varphi}\left(\boldsymbol{d}(\varphi)\otimes\boldsymbol{d}(\vartheta)\right)\right] \\
&= \boldsymbol{\mathcal{G}}\left[\frac{\partial}{\partial\varphi}\left(\boldsymbol{d}(\varphi)\right)\otimes\boldsymbol{d}(\vartheta) + \boldsymbol{d}(\varphi)\otimes\frac{\partial}{\partial\varphi}\left(\boldsymbol{d}(\vartheta)\right)\right] \\
&= \boldsymbol{\mathcal{G}}\left[\frac{\partial}{\partial\varphi}\left(\boldsymbol{d}(\varphi)\right)\otimes\boldsymbol{d}(\vartheta)\right],
\end{aligned} \tag{2.21}$$

where

$$\left[\frac{\partial}{\partial\vartheta}\left(\boldsymbol{d}(\vartheta)\right)\right]_{m_e} = \frac{jm_e}{\sqrt{2\pi}}e^{jm_e\vartheta}, \quad m_e = \ldots,-1,0,1,\ldots, \tag{2.22}$$

and

$$\left[\frac{\partial}{\partial\varphi}\left(\boldsymbol{d}(\varphi)\right)\right]_{m_a} = \frac{jm_a}{\sqrt{2\pi}}e^{jm_a\varphi}, \quad m_a = \ldots,-1,0,1,\ldots. \tag{2.23}$$

2.2 Classical Estimation Theory

In classical estimation, in contrast to Bayesian estimation, the parameters to be estimated are considered unknown and deterministic. In this thesis, we seek to estimate a set of unknown parameters, $\boldsymbol{\Omega}$, that are used to parameterize the steering vector $\boldsymbol{a}(\boldsymbol{\Omega})$. These parameters will then provide us with, either directly or indirectly, information about the DOA of the impinging waveform, or the position of the UE.

In the sections that follow, we present the maximum-likelihood (ML) method for parame-

ter estimation. Assuming that the impinging waveform $s(t)$ is deterministic, we derive an estimator for the unknown set of parameters, formulated as an optimization problem.

2.2.1 Maximum-Likelihood Estimation

In classical estimation theory the main principle is that we desire to estimate a set of unknown, deterministic parameters given observations, e.g. we want to estimate the direction of arrival (DOA) of a waveform impinging on an antenna array, given observations of the array output.

In this thesis, the observations are snapshots of the array output at times t_1 through t_N, and the set of parameters is e.g the DOA (if far field is assumed) of the impinging signal. Snapshots at different points in time are assumed to be independent due to the noise assumptions of temporal and spatial whiteness. This means that the likelihood of a collection of snapshots is the product of the likelihoods of all the individual snapshots, i.e.

$$f(\boldsymbol{x}(t_1),\ldots,\boldsymbol{x}(t_N) \mid \boldsymbol{\Omega},\boldsymbol{\xi},\sigma^2) = f(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega},\boldsymbol{\xi},\sigma^2) = \prod_{i=1}^{N} f(\boldsymbol{x}(t_i) \mid \boldsymbol{\Omega},\boldsymbol{\xi},\sigma^2). \quad (2.24)$$

The notation $\boldsymbol{x}(\boldsymbol{t}) \triangleq \{\boldsymbol{x}(t_1),\ldots,\boldsymbol{x}(t_N)\}$ is introduced in order to simplify notation, and will be used throughout the thesis. Note that f is used as notation for *both* the joint PDF and the PDF for a single array output. They are distinguished by the number of arguments provided to that function. The set of unknown parameters $\boldsymbol{\Omega}$ is then estimated as

$$\begin{aligned}\hat{\boldsymbol{\Omega}} &= \arg\max_{\boldsymbol{\Omega}} \left(\max_{\boldsymbol{\xi},\sigma^2} f(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega},\boldsymbol{\xi},\sigma^2) \right) \\ &= \arg\max_{\boldsymbol{\Omega}} \left(\max_{\boldsymbol{\xi},\sigma^2} \prod_{i=1}^{N} f(\boldsymbol{x}(t_i) \mid \boldsymbol{\Omega},\boldsymbol{\xi},\sigma^2) \right).\end{aligned} \quad (2.25)$$

Throughout this thesis we let $\boldsymbol{\xi}$ denote a set of nuisance parameters, i.e. the unknown signal waveform $s(t) \in \mathbb{C}$.

All this means is that, in maximum-likelihood (ML) estimation, we estimate the set of parameters as the parameter-values that makes the observations made as probable as possible [3].

2.2.2 Deterministic Maximum-Likelihood Estimation

In sensor array signal processing two different forms of the maximum-likelihood method are used, the deterministic maximum-likelihood (DML) method, and the stochastic maximum-likelihood (SML) method. The only difference between these two methods concerns the assumptions made about the incoming signal waveform, $s(t)$. In the DML estimation, $s(t)$ is assumed unknown and deterministic, while in SML estimation $s(t)$ is considered stochastic [12]. In this thesis, $s(t)$ is assumed to be deterministic, thus we only use DML for parameter estimation.

Under the assumption that $s(t)$ is deterministic, the output of the array at time t_i, given in (2.5), is simply an L-dimensional stochastic variable whose distribution is given by

$$\boldsymbol{x}(t_i) \sim \mathcal{CN}(\boldsymbol{a}(\boldsymbol{\Omega})s(t_i), \sigma^2\boldsymbol{I}_{L\times L}). \quad (2.26)$$

The noise is assumed to be spatially and temporally white, so we can write the likelihood of observing $\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_N)$ jointly as the product of the likelihoods of observing $\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_N)$ individually, as given in (2.24). Using (2.26), and letting $\boldsymbol{\xi} = \{s(t)\}$, (2.24) can be rewritten as

$$f(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega}, s(t), \sigma^2) = \prod_{i=1}^{N} \frac{1}{(\pi\sigma^2)^L} e^{-||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2/\sigma^2}, \tag{2.27}$$

and (2.25) as

$$\begin{aligned} \hat{\boldsymbol{\Omega}} &= \arg\max_{\boldsymbol{\Omega}} \left(\max_{s(t),\sigma^2} f(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega}, s(t), \sigma^2) \right) \\ &= \arg\max_{\boldsymbol{\Omega}} \left(\max_{s(t),\sigma^2} \prod_{i=1}^{N} \frac{1}{(\pi\sigma^2)^L} e^{-||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2/\sigma^2} \right). \end{aligned} \tag{2.28}$$

Instead of solving the maximization problem posed in (2.28), we will solve the equivalent problem of minimizing the negative log-likelihood function, where the log-likelihood function $\mathcal{L}$ is defined by

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega}, s(t), \sigma^2) &= \ln\left(f(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega}, s(t), \sigma^2)\right) \\ &= \ln\left(\prod_{i=1}^{N} \frac{1}{(\pi\sigma^2)^L} e^{-||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2/\sigma^2} \right) \\ &= -NL\ln(\pi) - NL\ln(\sigma^2) \\ &\quad - \frac{1}{\sigma^2} \sum_{i=1}^{N} ||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2. \end{aligned} \tag{2.29}$$

The first term in (2.29) is constant and does not affect the minimization, thus we have that the DML-estimate of $\boldsymbol{\Omega}$ becomes

$$\begin{aligned} \hat{\boldsymbol{\Omega}} &= \arg\min_{\boldsymbol{\Omega}} \left(\min_{s(t),\sigma^2} \left(-\mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Omega}, s(t), \sigma^2)\right) \right) \\ &= \arg\min_{\boldsymbol{\Omega}} \left(\min_{s(t),\sigma^2} \left(NL\ln(\sigma^2) + \frac{1}{\sigma^2} \sum_{i=1}^{N} ||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2 \right) \right) \\ &= \arg\min_{\boldsymbol{\Omega}} \left(\min_{s(t),\sigma^2} \left(L\ln(\sigma^2) + \frac{1}{N\sigma^2} \sum_{i=1}^{N} ||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2 \right) \right), \end{aligned} \tag{2.30}$$

where division by N in the last equality does not affect the optimum, since N is a positive constant. Division by N done in order to obtain a more convenient expression for the DML estimator.

In order to identify the value of $\boldsymbol{\Omega}$ that maximizes the objective function, we first identify the ML estimates of $s(t)$ and σ^2, starting with $s(t)$. From linear algebra, we know that

$\boldsymbol{x} = \boldsymbol{A}^\dagger \boldsymbol{y}$ provides the least-squares solution to the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$, thus

$$||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2 \tag{2.31}$$

is minimized by

$$\hat{s}(t_i) = \boldsymbol{a}(\boldsymbol{\Omega})^\dagger \boldsymbol{x}(t_i), \tag{2.32}$$

which then is the ML-estimate of $s(t_i)$. From linear algebra, we also have that

$$\boldsymbol{a}(\boldsymbol{\Omega})\boldsymbol{a}(\boldsymbol{\Omega})^\dagger \boldsymbol{x}(t_i)$$

is the orthogonal projection of $\boldsymbol{x}(t_i)$ onto the column space of $\boldsymbol{a}(\boldsymbol{\Omega})$. By substituting $s(t_i)$ in (2.31) for $\hat{s}(t_i) = \boldsymbol{a}(\boldsymbol{\Omega})^\dagger \boldsymbol{x}(t_i)$, we obtain

$$\begin{aligned} ||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})s(t_i)||^2 &= ||\boldsymbol{x}(t_i) - \boldsymbol{a}(\boldsymbol{\Omega})\boldsymbol{a}(\boldsymbol{\Omega})^\dagger \boldsymbol{x}(t_i)||^2 \\ &= ||\boldsymbol{x}(t_i) - \boldsymbol{\Pi}_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)||^2, \end{aligned} \tag{2.33}$$

where $\boldsymbol{\Pi}_{\boldsymbol{a}(\boldsymbol{\Omega})}$ denotes orthogonal projection onto the columns of $\boldsymbol{a}(\boldsymbol{\Omega})$. Finally, (2.33) is simplified by noting that

$$\boldsymbol{x}(t_i) - \boldsymbol{\Pi}_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i) = \boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i), \tag{2.34}$$

where $\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}$ denotes the orthogonal projection onto the orthogonal complement of the columns of $\boldsymbol{a}(\boldsymbol{\Omega})$. Using this result, we can rewrite the term containing the sum in (2.30) as

$$\begin{aligned} \frac{1}{N\sigma^2}\sum_{k=1}^{N} ||\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)||^2 &\overset{\text{(a)}}{=} \frac{1}{N\sigma^2}\sum_{k=1}^{N} \text{tr}((\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i))^H \boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)) \\ &\overset{\text{(b)}}{=} \frac{1}{N\sigma^2}\sum_{k=1}^{N} \text{tr}(\boldsymbol{x}(t_i)^H (\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})})^H \boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)) \\ &\overset{\text{(c)}}{=} \frac{1}{N\sigma^2}\sum_{k=1}^{N} \text{tr}(\boldsymbol{x}(t_i)^H \boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)) \\ &\overset{\text{(d)}}{=} \frac{1}{N\sigma^2}\sum_{k=1}^{N} \text{tr}(\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{x}(t_i)\boldsymbol{x}(t_i)^H) \\ &\overset{\text{(e)}}{=} \frac{1}{\sigma^2}\text{tr}(\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}(t_i)\boldsymbol{x}(t_i)^H) \\ &\overset{\text{(f)}}{=} \frac{1}{\sigma^2}\text{tr}(\boldsymbol{\Pi}^\perp_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}), \end{aligned} \tag{2.35}$$

where the (2.35.a) is due to the fact that

$$||\boldsymbol{u}||^2 = \boldsymbol{u}^H\boldsymbol{u} = \text{tr}(\boldsymbol{u}^H\boldsymbol{u}).$$

Moreover, (2.35.b) is due to the properties of the Hermitian transpose, (2.35.c) holds because orthogonal projections are Hermitian and idempotent, (2.35.d) holds due to the cyclic properties of the trace-operator, and (2.35.e) holds since the trace operator is linear.

In (2.35.f) we define the sample covariance $\hat{\boldsymbol{R}}$ by

$$\hat{\boldsymbol{R}} \triangleq \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}(t_i)\boldsymbol{x}(t_i)^H. \tag{2.36}$$

We now seek to obtain the ML estimate of σ^2. Using the result from (2.35) in (2.30) we get

$$\hat{\boldsymbol{\Omega}} = \arg\min_{\boldsymbol{\Omega}} \left(\min_{\sigma^2} \left(L\ln(\sigma^2) + \frac{1}{\sigma^2}\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}) \right) \right). \tag{2.37}$$

We find the ML-estimate of σ^2 by differentiating the objective function in (2.37) with respect to σ^2, keeping $\boldsymbol{\Omega}$ fixed, and equating the derivative to zero. We get

$$\frac{\partial}{\partial(\sigma^2)} \left(L\ln(\sigma^2) + \frac{1}{\sigma^2}\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}) \right) = \frac{L}{\sigma^2} - \frac{1}{\sigma^4}\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}) = 0. \tag{2.38}$$

Solving (2.38) for σ^2 then yields the ML estimate of σ^2 as

$$\hat{\sigma}^2 = \frac{1}{L}\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}). \tag{2.39}$$

Using the results from (2.39) in (2.37), we obtain

$$\hat{\boldsymbol{\Omega}} = \arg\min_{\boldsymbol{\Omega}} \left(L\left(\ln\left(\frac{1}{L}\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}) \right) + 1 \right) \right), \tag{2.40}$$

which is equivalent to

$$\hat{\boldsymbol{\Omega}} = \arg\min_{\boldsymbol{\Omega}} \left(\text{tr}(\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\hat{\boldsymbol{R}}) \right). \tag{2.41}$$

This is a derivation of the result found in e.g. [12].

2.2.3 The Cramér-Rao Bound

The Cramér-Rao bound (CRB) is a lower bound on the mean square error (MSE) of any unbiased parameter estimator $\hat{\boldsymbol{\Omega}}$ of $\boldsymbol{\Omega}$. An estimator is said to be unbiased if it satisfies

$$\mathbb{E}\{\hat{\boldsymbol{\Omega}}\} - \boldsymbol{\Omega} = \boldsymbol{0}.$$

Let $\boldsymbol{C}_{\hat{\boldsymbol{\Omega}}}$ be the covariance of $\hat{\boldsymbol{\Omega}}$. The CRB then states that the following inequality holds

$$\boldsymbol{C}_{\hat{\boldsymbol{\Omega}}} = \mathbb{E}\{(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega})(\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega})^T\} \succeq \text{CRB}_{\hat{\boldsymbol{\Omega}}}, \tag{2.42}$$

where the notation $\boldsymbol{A} \succeq \boldsymbol{B}$ means that $\boldsymbol{A} - \boldsymbol{B}$ is positive semi definite [18], and $\text{CRB}_{\hat{\boldsymbol{\Omega}}}$ is the CRB for $\boldsymbol{\Omega}$.

In this section we derive the CRB for the DML estimate of $\boldsymbol{\Omega}$, given one single snapshot of the antenna array output, i.e. $N = 1$. This is done without loss of generality, since information is additive [9]. The derivation and notation will be based on [18].

2.2.4 Derivation of the CRB for the Used Array Model

The CRB is given as the inverse of the Fisher Information Matrix (FIM) [9]. Thus, in order to derive an expression for the CRB, we first need to derive an expression for the FIM. Note that, even though we are not interested in knowing the nuisance parameters $\boldsymbol{\xi}$ we need to include them in our calculations of the FIM, since they affect the CRB of $\boldsymbol{\Omega}$. Let $n_{\boldsymbol{\Omega}}$ and n_{ξ} be the cardinality of $\boldsymbol{\Omega}$ and $\boldsymbol{\xi}$ respectively. We introduce the set

$$\boldsymbol{\Psi} = \{\psi_1, \ldots, \psi_{n_{\boldsymbol{\Omega}}+n_{\xi}}\} \triangleq \boldsymbol{\Omega} \cup \boldsymbol{\xi} = \{\omega_1, \ldots \omega_{n_{\boldsymbol{\Omega}}}, \xi_1, \ldots, \xi_{n_{\xi}}\}, \tag{2.43}$$

to simplify notation when deriving the FIM. We can now write the output of the array given in (2.5) as

$$\boldsymbol{x}(t) = \boldsymbol{u}(\boldsymbol{\Psi}) + \boldsymbol{n}(t) \sim \mathcal{CN}(\boldsymbol{u}(\boldsymbol{\Psi}), \sigma^2 \boldsymbol{I}_{L\times L}), \tag{2.44}$$

where $\boldsymbol{u}(\boldsymbol{\Psi}) = \boldsymbol{a}(\boldsymbol{\Omega})s(t)$.

What we want to derive is an expression for the FIM of the parameters $\boldsymbol{\Psi}$, $\mathcal{I}(\boldsymbol{\Psi}, \sigma^2)$.

The FIM of the parameters is defined as

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\Psi}, \sigma^2) = -\mathbb{E}\left\{ \left(\nabla_{\psi} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2)\right) \left(\nabla_{\psi} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2)\right)^T \right\}, \tag{2.45}$$

where

$$\nabla_{\psi} = \begin{bmatrix} \frac{\partial}{\partial \psi_1} & \cdots & \frac{\partial}{\partial \psi_{n_{\boldsymbol{\Omega}}+n_{\xi}}} \end{bmatrix}^T,$$

and $\mathcal{L}$ is the same log-likelihood as defined in (2.29), where $\boldsymbol{\Omega}$ and $\boldsymbol{\xi}$ have been replaced by the equivalent $\boldsymbol{\Psi}$. The element on row k and in column j in the FIM is given by [18]

$$\boldsymbol{\mathcal{I}}_{k,j}(\boldsymbol{\Psi}, \sigma^2) = -\mathbb{E}\left\{ \frac{\partial^2}{\partial \psi_i \partial \psi_j} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2) \right\}. \tag{2.46}$$

To find (2.46), we start by computing

$$\frac{\partial}{\partial \psi_j} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2).$$

By direct calculation, we obtain:

$$\begin{aligned} &\frac{\partial}{\partial \psi_j} \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2) \\ &= \frac{\partial}{\partial \psi_j} \left(-NL\ln(\pi) - NL\ln(\sigma^2) - \frac{1}{\sigma^2} \|\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\|^2 \right) \\ &= \frac{\partial}{\partial \psi_j} \left(-\frac{1}{\sigma^2} \|\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\|^2 \right), \end{aligned} \tag{2.47}$$

which, by using the relation

$$\|\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\|^2 = \left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right)^H \left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right),$$

can be rewritten as

$$\begin{aligned}
&= \frac{1}{\sigma^2}\Bigg(\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\boldsymbol{x}(t_i) - \frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\boldsymbol{u}(\boldsymbol{\Psi}) \\
&\quad + \boldsymbol{x}^H(t_i)\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right) - \boldsymbol{u}^H(\boldsymbol{\Psi})\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\Bigg) \\
&= \frac{1}{\sigma^2}\Bigg(\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right) + \left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right)^H\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\Bigg).
\end{aligned} \tag{2.48}$$

Using the relation $z + z^* = 2\Re\{z\}$ for $z \in \mathbb{C}$, where z^* denotes complex conjugate, we then obtain

$$\frac{\partial}{\partial\psi_j}\mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2) = \frac{2}{\sigma^2}\Re\left\{\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right)\right\}. \tag{2.49}$$

By differentiating (2.49) with respect to ψ_k, we obtain

$$\begin{aligned}
\frac{\partial^2}{\partial\psi_k\partial\psi_j}\mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2) = \frac{2}{\sigma^2}\Re\Bigg\{&\frac{\partial^2}{\partial\psi_k\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\left(\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\right) \\
&- \frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\frac{\partial}{\partial\psi_k}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\Bigg\}.
\end{aligned} \tag{2.50}$$

In order to compute the element on row k and column j of the FIM for the parameters, we need to compute the negative expectation of (2.50). Note that

$$\mathbb{E}\{\boldsymbol{x}(t_i) - \boldsymbol{u}(\boldsymbol{\Psi})\} = 0.$$

Thus, we have that the element on row k and column j of the FIM for the parameters $\boldsymbol{\Psi}$ is given by

$$\begin{aligned}
-\mathbb{E}\left\{\frac{\partial^2}{\partial\psi_k\partial\psi_j}\mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \boldsymbol{\Psi}, \sigma^2)\right\} &= \frac{2}{\sigma^2}\Re\left\{\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\frac{\partial}{\partial\psi_k}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\right\} \\
&= \frac{2}{\sigma^2}\Re\left\{\frac{\partial}{\partial\psi_k}\left(\boldsymbol{u}^H(\boldsymbol{\Psi})\right)\frac{\partial}{\partial\psi_j}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\right\},
\end{aligned} \tag{2.51}$$

where the last equality holds because a real number is equal to its complex conjugate. We define the matrix $\boldsymbol{D}(\boldsymbol{\Psi})$ as

$$\boldsymbol{D}(\boldsymbol{\Psi}) = \left[\frac{\partial}{\partial\psi_1}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right) \quad \cdots \quad \frac{\partial}{\partial\psi_{n_{\boldsymbol{\Omega}}+n_\xi}}\left(\boldsymbol{u}(\boldsymbol{\Psi})\right)\right]. \tag{2.52}$$

This definition enables us to write the FIM for the parameters in $\boldsymbol{\Psi}$ compactly as

$$\boldsymbol{\mathcal{I}}(\boldsymbol{\Psi}, \sigma^2) = \frac{2}{\sigma^2}\Re\left\{\boldsymbol{D}^H(\boldsymbol{\Psi})\boldsymbol{D}(\boldsymbol{\Psi})\right\}. \tag{2.53}$$

From the full expression for the FIM given in (2.53), we obtain the CRB for the parameters

in $\mathbf{\Psi}$ as

$$\text{CRB}_{\mathbf{\Psi}} = \boldsymbol{\mathcal{I}}(\mathbf{\Psi}, \sigma^2)^{-1}, \tag{2.54}$$

and the CRB for $\mathbf{\Omega}$, the parameters of interest, as the $n_{\mathbf{\Omega}} \times n_{\mathbf{\Omega}}$-dimensional top-left block of $\text{CRB}_{\mathbf{\Psi}}$.

2.3 Optimization

In order to find the unknown parameters $\mathbf{\Omega}$, e.g. by the DML estimator derived in section 2.2.2, an optimization problem needs to be solved. The previously derived estimator, given in (2.41) is a problem of a highly nonlinear nature, making it necessary for us to utilize nonlinear optimization methods in order to solve it. In this section, we present some methods for non-linear optimization that will later be used in order to obtain parameter estimates.

Most numerical optimization methods are iterative, i.e. the estimates of the sought after unknown parameters $\mathbf{\Omega}$ at iteration $k+1$ are dependent on the estimates at iteration k according to

$$\hat{\mathbf{\Omega}}_{k+1} = \hat{\mathbf{\Omega}}_k + \mu_k \boldsymbol{\zeta}_k, \tag{2.55}$$

where μ_k is the step size and $\boldsymbol{\zeta}_k$ is the search direction at iteration k [9]. A few such algorithms will be described in the following sections. Algorithms of this kind must be provided with an initial estimate of the unknown parameters, $\hat{\mathbf{\Omega}}_0$.

Using the algorithms described below, we will perform minimization of a general objective-function $\mathcal{C}(\mathbf{\Omega})$. For some of the algorithms, we will also discuss the special case most interesting to us: minimization of the negative log-likelihood function, where the log-likelihood function was defined in (2.29).

2.3.1 The Grid Search Method

The initial parameter estimates used by the optimization algorithms in this thesis are found by the grid search method. In the grid search method we sample the objective function at a set of discrete points and then estimate $\hat{\mathbf{\Omega}}_0$ as the point that yielded the most optimal value of the objective function.

Assume that we sample the objective function uniformly for n different values of each parameter. The time complexity of the algorithm is then $\mathcal{O}(n^k)$, where k is the number of parameters, i.e. the algorithm is not computationally efficient. This is why the objective function is sampled rather sparsely, and used only to provide an initial estimate.

2.3.2 The Steepest Descent Algorithm

In the steepest-descent method, we choose the search direction in each iteration as the negative gradient of the objective function [14], i.e.

$$\hat{\mathbf{\Omega}}_{k+1} = \hat{\mathbf{\Omega}}_k - \mu_k \nabla_\omega \mathcal{C}(\mathbf{\Omega}), \tag{2.56}$$

where $\nabla_\omega \mathcal{C}$ denotes the gradient of $\mathcal{C}$ with respect to the parameters in $\mathbf{\Omega}$. The negative gradient points in the direction of the parameter space in which the objective function is decreasing most rapidly.

In theory, assuming infinite computational precision, the steepest descent algorithm converges to a local minimum. One advantage with the algorithm is its low computational complexity [18]. One drawback of the algorithm is that, even though it might converge fast in some special cases, it can often converge slowly [14].

2.3.3 The Newton Algorithm

Consider the Taylor expansion about $\mathbf{\Omega} = \mathbf{\Omega}_k$ of the objective function $\mathcal{C}$, neglecting terms of order higher than two, i.e.

$$\begin{aligned}\mathcal{C}(\mathbf{\Omega}) \approx \mathcal{C}(\mathbf{\Omega}_k) + (\mathbf{\Omega} - \mathbf{\Omega}_k)^T \nabla_\omega \mathcal{C}(\mathbf{\Omega}_k) \\ + \frac{1}{2}(\mathbf{\Omega} - \mathbf{\Omega}_k)^T \left(\nabla_\omega \nabla_\omega^T \mathcal{C}(\mathbf{\Omega}_k)\right)(\mathbf{\Omega} - \mathbf{\Omega}_k)\end{aligned} \tag{2.57}$$

Using the expansion in (2.57) we obtain the Taylor expansion of the gradient:

$$\nabla_\omega \mathcal{C}(\mathbf{\Omega}) \approx \nabla_\omega \mathcal{C}(\mathbf{\Omega}_k) + \nabla_\omega \nabla_\omega^T \mathcal{C}(\mathbf{\Omega}_k)(\mathbf{\Omega} - \mathbf{\Omega}_k). \tag{2.58}$$

The objective function $\mathcal{C}$ is assumed to be quadratic between $\mathbf{\Omega}_k$ and the solution, $\mathbf{\Omega}_{k+1}$, which is a stationary point since the solution is the closest stationary point to the initial estimate $\hat{\mathbf{\Omega}}_0$, in our case a local minima of $\mathcal{C}$. Using this, alongside the result from (2.58) with $\mathbf{\Omega} = \mathbf{\Omega}_{k+1}$ we obtain

$$0 = \nabla_\omega \mathcal{C}(\mathbf{\Omega}_k) + \nabla_\omega \nabla_\omega^T \mathcal{C}(\mathbf{\Omega}_k)(\mathbf{\Omega}_{k+1} - \mathbf{\Omega}_k), \tag{2.59}$$

which can be rearranged on the form

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k - \left(\nabla_\omega \nabla_\omega^T \mathcal{C}(\mathbf{\Omega}_k)\right)^{-1} \nabla_\omega \mathcal{C}(\mathbf{\Omega}_k), \tag{2.60}$$

which is the update step for the Newton algorithm. Sometimes a (reduced) step-size μ_k is introduced in order to improve convergence. The purpose of doing so is to constrain the algorithm to only taking steps in a region where the quadratic assumption is good [18]. The modified method is called the *damped Newton method* and is given by [21], [18]:

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k - \mu_k \left(\nabla_\omega \nabla_\omega^T \mathcal{C}(\mathbf{\Omega}_k)\right)^{-1} \nabla_\omega \mathcal{C}(\mathbf{\Omega}_k). \tag{2.61}$$

Consider the special case where the objective function $\mathcal{C}$ is the log-likelihood function $\mathcal{L}$. The negative Hessian

$$-\nabla_\omega \nabla_\omega^T \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \mathbf{\Omega}_k, s(t), \sigma^2)$$

is called the Stochastic Fisher Information, and is denoted by [18]

$$\widetilde{\boldsymbol{\mathcal{I}}}(\mathbf{\Omega}_k, s(t), \sigma^2).$$

Thus, we can write (2.60) as

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k + \mu_k \widetilde{\boldsymbol{\mathcal{I}}}(\mathbf{\Omega}_k, s(t), \sigma^2)^{-1} \nabla_\omega \mathcal{L}(\boldsymbol{x}(\boldsymbol{t}) \mid \mathbf{\Omega}_k, s(t), \sigma^2), \tag{2.62}$$

which then is the Newton step for iterative ML estimation [18].

2.3.4 The Fisher-Scoring Method

In practice, to simplify the computational complexity, $\widetilde{\boldsymbol{\mathcal{I}}}(\boldsymbol{\Omega}_k, s(t), \sigma^2)$ in (2.62) is sometimes replaced with its expected value. By definition, we have that

$$\mathbb{E}\{\widetilde{\boldsymbol{\mathcal{I}}}(\boldsymbol{\Omega}_k, s(t), \sigma^2)\} = \boldsymbol{\mathcal{I}}(\boldsymbol{\Omega}_k, s(t), \sigma^2),$$

which is the Fisher information for $\boldsymbol{\Omega}$. There is also another advantage with replacing the stochastic Fisher information matrix with the FIM. In particular, the stochastic Fisher information matrix is not necessarily positive definite (hence it is not necessarily invertible), which is a property that the FIM has [18], thus the inverse of the FIM can always be found.

The simplified algorithm becomes

$$\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k + \boldsymbol{\mathcal{I}}(\boldsymbol{\Omega}_k, s(t), \sigma^2)^{-1} \nabla_\omega \mathcal{L}(\boldsymbol{x(t)} \mid \boldsymbol{\Omega}_k, s(t), \sigma^2). \tag{2.63}$$

This modified Newton algorithm is known as *The Fisher Scoring Method* [19] or The Gauss-Newton Method [18].

2.3.5 The Levenberg-Marquardt Method

When deriving the Newton method, it was assumed that the objective function is quadratic near $\boldsymbol{\Omega}_k$, which for non-linear functions might only hold approximately for a small region around $\boldsymbol{\Omega}_k$. The Levenberg-Marquardt Method aims to limit the step-size to the extent that the quadratic assumption holds [18].

The method can be expressed as [18]

$$\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k - \left(\nabla_\omega \nabla_\omega^T \mathcal{C}(\boldsymbol{\Omega}_k) + \mu_k \boldsymbol{I} \odot \nabla_\omega \nabla_\omega^T \mathcal{C}(\boldsymbol{\Omega}_k)\right)^{-1} \nabla_\omega \mathcal{C}(\boldsymbol{\Omega}_k), \tag{2.64}$$

where $\odot$ denotes the Schur product. Note that this update step is very similar to the update step of the Newton method, where the only difference is the term $\mu_k \boldsymbol{I} \odot \nabla_\omega \nabla_\omega^T \mathcal{C}(\boldsymbol{\Omega}_k)$ being added to the Hessian.

The Levenberg-Marquardt method can be seen as a combination of Newtons method and the steepest descent method. If $\mu_k \ll 1$, then the update step is the normal Newton method update step, as given in (2.60). If instead $\mu_k \gg 1$, then the update step becomes

$$\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k - \left(\mu_k \boldsymbol{I} \odot \nabla_\omega \nabla_\omega^T \mathcal{C}(\boldsymbol{\Omega}_k)\right)^{-1} \nabla_\omega \mathcal{C}(\boldsymbol{\Omega}_k), \tag{2.65}$$

which is the steepest descent update step with a so called matched metric [18].

If we approximate the Hessian as the FIM, like in Fisher scoring, the update-step becomes

$$\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k + \left(\boldsymbol{\mathcal{I}}(\boldsymbol{\Omega}_k, s(t), \sigma^2) + \mu_k \boldsymbol{I} \odot \boldsymbol{\mathcal{I}}(\boldsymbol{\Omega}_k, s(t), \sigma^2)\right)^{-1} \nabla_\omega \mathcal{L}(\boldsymbol{x(t)} \mid \boldsymbol{\Omega}_k, s(t), \sigma^2), \tag{2.66}$$

which is the method we will mainly use for optimization throughout this thesis.

2.4 Bayesian Filtering

Filtering can be seen as an extension of estimation, where we allow the parameter to change over time, according to some dynamic model [9]. In this work, we want to study

how the DOAs change over time as a transmitter is moving. We consider a non-linear state space model of the system with additive noise, which is given by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{v}_k \\ \boldsymbol{y}_k &= \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{e}_k, \end{aligned} \tag{2.67}$$

where $\boldsymbol{x}_k$ is the state, $\boldsymbol{y}_k$ is the measurement, $\boldsymbol{v}_k$ and $\boldsymbol{e}_k$ are noise terms, all at time k, and $\boldsymbol{f}$ and $\boldsymbol{h}$ are two, possibly non-linear, functions, which in general are vector-valued, hence the use of boldface notation.

2.4.1 The Kalman Filter

In the special case where $\boldsymbol{f}$ and $\boldsymbol{h}$ are linear functions, both the motion model and the measurement model will be linear, i.e. $\boldsymbol{f}$ and $\boldsymbol{h}$ are simply linear transformations, and the linear state space model of the system is then given by

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{G}\boldsymbol{v}_k \\ \boldsymbol{y}_k &= \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{e}_k, \end{aligned} \tag{2.68}$$

where $\boldsymbol{F}$ and $\boldsymbol{H}$ are matrices, describing the motion and measurement respectively. For this system, assuming that $\boldsymbol{v}_k$ and $\boldsymbol{e}_k$ are Gaussian, with covariances $\text{Cov}(\boldsymbol{v}_k) = \mathbb{E}\{\boldsymbol{v}_k \boldsymbol{v}_k^T\} = \boldsymbol{Q}$ and $\text{Cov}(\boldsymbol{e}_k) = \mathbb{E}\{\boldsymbol{e}_k \boldsymbol{e}_k^T\} = \boldsymbol{R}$, then the best linear unbiased filter is given by the Kalman filter (KF) [9]. The Kalman filter was first proposed in 1960 by R. Kalman in [10], providing a more practical solution to the Wiener problem, by operating only in the time domain, as opposed to operating in the spectral domain. The KF is a recursive algorithm that consists of two steps. The steps are

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{\varepsilon}_k \\ \boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{H} \boldsymbol{P}_{k|k-1}, \end{aligned} \tag{2.69}$$

and

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{F}\hat{\boldsymbol{x}}_{k|k} \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^T, \end{aligned} \tag{2.70}$$

where

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^T \left(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T + \boldsymbol{R}\right)^{-1}, \tag{2.71}$$

and

$$\boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}. \tag{2.72}$$

The indices of the state estimate, e.g. $\hat{\boldsymbol{x}}_{k|k}$ and $\hat{\boldsymbol{x}}_{k|k-1}$, should be interpreted as "the estimated state at time k given observations up to, and including, time k" and "the estimated state at time k given observations up to, and including, time $k-1$" respectively. The indices of the state estimate covariances follow the same convention. $\boldsymbol{K}_k$ is called the Kalman gain, and $\boldsymbol{\varepsilon}_k$ is called the innovation. Equation 2.69 is called the measurement update, and equation 2.70 is known as the time update of the Kalman filter. The filter is initialized with $\hat{\boldsymbol{x}}_{1|0} = \boldsymbol{x}_0$ and $\boldsymbol{P}_{1|0} = \boldsymbol{P}_0$, which are chosen based on an initial state estimate. More in depth information about the Kalman filter is given in e.g. [9].

2.4.2 The Extended Kalman Filter

The Kalman filter, as described in the previous section, only provides a state estimation algorithm for linear models. Thus, it is not applicable for general, non-linear, systems. However, it is common to approximate the non-linear system of interest as linear by linearizing $\boldsymbol{f}$ and $\boldsymbol{h}$ around the current state estimate using a Taylor expansion. From this, we obtain measurement and time update steps similar to those of the standard Kalman filter. This is refered to as the Extended Kalman Filter (EKF). In contrast to the KF, we now also have to perform linearization in each step of the recursion. The measurement- and time-update steps of the recursion for the EKF are given by [9]

$$\begin{aligned}\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{\varepsilon}_k \\ \boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{P}_{k|k-1}(\boldsymbol{h}'(\hat{\boldsymbol{x}}_{k|k-1}))^T \boldsymbol{S}_k^{-1} \boldsymbol{h}'(\hat{\boldsymbol{x}}_{k|k-1}) \boldsymbol{P}_{k|k-1}\end{aligned} \tag{2.73}$$

and

$$\begin{aligned}\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{f}(\hat{\boldsymbol{x}}_{k|k}) \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{Q} + \boldsymbol{f}'(\hat{\boldsymbol{x}}_{k|k}) \boldsymbol{P}_{k|k} (\boldsymbol{f}'(\hat{\boldsymbol{x}}_{k|k}))^T,\end{aligned} \tag{2.74}$$

respectively. Here $\boldsymbol{S}_k$, $\boldsymbol{K}_k$, and $\boldsymbol{\varepsilon}_k$ are given by

$$\boldsymbol{S}_k = \boldsymbol{R} + \boldsymbol{h}'(\hat{\boldsymbol{x}}_{k|k-1}) \boldsymbol{P}_{k|k-1} (\boldsymbol{h}'(\hat{\boldsymbol{x}}_{k|k-1}))^T, \tag{2.75}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k+1|k} (\boldsymbol{h}'(\hat{\boldsymbol{x}}_{k|k-1}))^T \boldsymbol{S}_k^{-1}, \tag{2.76}$$

and

$$\boldsymbol{\varepsilon}_k = \boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}), \tag{2.77}$$

respectively. Additionally, $\boldsymbol{f}'$ and $\boldsymbol{h}'$ denotes the Jacobians with respect to the state of $\boldsymbol{f}$ and $\boldsymbol{h}$, respectively.

2.4.3 An Alternative Form of the EKF

In this work we will not use the standard EKF as presented in [9], but rather the EKF given in [13] and [20]. The reason for using this alternative form is motivated by computational complexity. For a 100 antenna array, in the standard EKF we need to invert a 100×100 matrix, while for the alternative form we only need to invert a $p \times p$ matrix, where p is the number of states in the model. In this thesis, we have that $p = 6$. Even though we assume a linear motion model, we cannot use standard KF, since the measurement model is still assumed to be non-linear.

In this form of the EKF, the measurement update is given by

$$\begin{aligned}\boldsymbol{P}_{k|k} &= \left(\left(\boldsymbol{P}_{k|k-1} \right)^{-1} + \boldsymbol{J}_k \right)^{-1} \\ \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{P}_{k|k-1} \boldsymbol{v}_k,\end{aligned} \tag{2.78}$$

and the time update is given by

$$\begin{aligned}\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{F}\hat{\boldsymbol{x}}_{k|k} \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{F}\boldsymbol{P}_{k|k}\boldsymbol{F}^T + \boldsymbol{Q}.\end{aligned} \tag{2.79}$$

In equation (2.78), $\boldsymbol{J}_k$ and $\boldsymbol{v}_k$ are the FIM and score function respectively, both evaluated at $\hat{\boldsymbol{x}}_{k|k-1}$.

The Fisher information $\boldsymbol{J}_k$ is computed, with the formula given in [21], as

$$\boldsymbol{J_k} = 2\Re\{(\boldsymbol{D}^H\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{D}) \odot (\boldsymbol{w}^T\boldsymbol{a}(\boldsymbol{\Omega})^{\dagger}\hat{\boldsymbol{R}}(\boldsymbol{a}(\boldsymbol{\Omega})^{\dagger})^H\boldsymbol{w})^T\}, \tag{2.80}$$

where $\boldsymbol{w} = \begin{bmatrix}1 & 1\end{bmatrix}$. Note that $\boldsymbol{J}_k$ is only the FIM for the two incident angles, ϑ and φ. Assume that we have m states other than ϑ and φ in our model. The FIM for all states is then given by

$$\boldsymbol{J}_k^{\text{full}} = \text{blkdiag}\left(\boldsymbol{J}_k, \boldsymbol{0}_{m\times m}\right), \tag{2.81}$$

since we cannot deduce any information about the other states from the array output. The score function $\boldsymbol{v}_k$ is computed, with the formula given in [21], as

$$\boldsymbol{v}_k = -2\Re\{\text{diag}(\boldsymbol{w}^T\boldsymbol{a}(\boldsymbol{\Omega})^{\dagger}\hat{\boldsymbol{R}}\boldsymbol{\Pi}^{\perp}_{\boldsymbol{a}(\boldsymbol{\Omega})}\boldsymbol{D})\}. \tag{2.82}$$

Again, this is not as general as the EKF given in section 2.4.2, since a linear motion model is assumed, which simplifies computations considerably. However, the linear *constant angular acceleration model*, presented in section 2.4.4, still proves to yield good results, despite only being an approximation of the true motion model.

2.4.4 Motion Models

In the previous sections, we have given an overview of different Kalman filter algorithms. These were given for arbitrary systems, without any assumption on the governing functions $\boldsymbol{f}$ and $\boldsymbol{g}$. In practice, having a model that provides a good description of the system dynamics at hand is of great importance [9], especially if we have high demands on accuracy, since a model that is a good approximation of the system will yield better filtering results than one that is not.

In this thesis, we will use simple linear models to describe the dynamics of the system. In the sections that follow, we will present three different motion models: the constant angle, the constant angular velocity model, and the constant angular acceleration model. These three motion models are all linear. The three models are closely related, and the only difference is how many derivatives of the incident angles ϑ and φ that are included in the state vector.

Constant Angle Model

The constant angle model is the first, and simplest, motion model we will use in this thesis. The model assumes that the angles to be estimated, ϑ and φ, remain constant through time, and that any change in these, driven by the process noise, is to be interpreted as a change in ϑ and φ. The model is given by

$$\begin{bmatrix}\vartheta_{k+1} \\ \varphi_{k+1}\end{bmatrix} = \boldsymbol{I}_{2\times 2}\begin{bmatrix}\vartheta_k \\ \varphi_k\end{bmatrix} + \tau\boldsymbol{I}_{2\times 2}\boldsymbol{w}, \tag{2.83}$$

where the process noise, $\boldsymbol{w}$, is assumed to be a two-dimensional real Gaussian random variable, where the covariance of $\boldsymbol{w}$ is assumed to be on the form $\sigma_w \boldsymbol{I}_{2\times 2}$. τ is the sample time.

Constant Angular Velocity Model

In the constant angular velocity model, the incident angles ϑ and φ, are assumed to be varying with a constant angular velocity. The process noise is to be interpreted as angular acceleration, change in the angular velocity. The model is given by

$$\begin{bmatrix} \vartheta_{k+1} \\ \varphi_{k+1} \\ \dot{\vartheta}_{k+1} \\ \dot{\varphi}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & \tau \boldsymbol{I}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \end{bmatrix} \begin{bmatrix} \vartheta_k \\ \varphi_k \\ \dot{\vartheta}_k \\ \dot{\varphi}_k \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} \boldsymbol{I}_{2\times 2} \\ \tau \boldsymbol{I}_{2\times 2} \end{bmatrix} \boldsymbol{w}, \tag{2.84}$$

where the process noise, $\boldsymbol{w}$, is assumed to be a two-dimensional real Gaussian random variable, where the covariance of $\boldsymbol{w}$ is assumed to be on the form $\sigma_w \boldsymbol{I}_{2\times 2}$. τ is the sample time.

Constant Angular Acceleration Model

In the constant angular acceleration model, the incident angles ϑ and φ are assumed to be varying with a constant angular acceleration. The process noise is to be interpreted as change in the angular acceleration. The model is given by

$$\begin{bmatrix} \vartheta_{k+1} \\ \varphi_{k+1} \\ \dot{\vartheta}_{k+1} \\ \dot{\varphi}_{k+1} \\ \ddot{\vartheta}_{k+1} \\ \ddot{\varphi}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & \tau \boldsymbol{I}_{2\times 2} & \frac{\tau^2}{2} \boldsymbol{I}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} & \tau \boldsymbol{I}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \end{bmatrix} \begin{bmatrix} \vartheta_k \\ \varphi_k \\ \dot{\vartheta}_k \\ \dot{\varphi}_k \\ \ddot{\vartheta}_k \\ \ddot{\varphi}_k \end{bmatrix} + \begin{bmatrix} \frac{\tau^3}{6} \boldsymbol{I}_{2\times 2} \\ \frac{\tau^2}{2} \boldsymbol{I}_{2\times 2} \\ \tau \boldsymbol{I}_{2\times 2} \end{bmatrix} \boldsymbol{w}, \tag{2.85}$$

where the process noise, $\boldsymbol{w}$, is assumed to be a two-dimensional real Gaussian random variable, with covariance of $\boldsymbol{w}$ is assumed to be on the form $\sigma_w \boldsymbol{I}_{2\times 2}$. τ is the sample time. The constant angular acceleration model is the motion model used in the EKFs throughout this thesis.

3
Simulation Setup

In this section we present simulation results generated in the same simulation environment as used in [13]. There are, however, a few key differences. In this work, we consider the case where the antennas of the antenna array observe different angles (i.e. no geometric far-field assumption). Also, we only use vertical dipoles (both at the UE and base station), i.e. we do not employ polarimetric antennas in the antenna array, which could possibly improve performance.

3.1 The Madrid Grid

The Madrid Grid refers to the METIS Madrid grid environmental model, is a model of a dense urban city. It is a compromise between a realistic urban setting, i.e. non-homogenous building layout, and existing models like the Manhattan grid [16].

The Madrid Grid represents the urban Madrid street grid. In particular interest to us is the Gran-Via, a large street in central Madrid. An overview of the Madrid Grid, where the Gran-Via is highlighted, is depicted in Figure 3.1.

3.2 Base Station

In all the simulations the same base station position will be used. This base station will be placed at the bottom of the Gran-Via at a height of 50 m.

All simulations are done using a URA of either 25 or 100 antennas in the base station. The antenna elements in the array are modeled as vertical dipoles.

At each antenna, noise is added according to the model given in equation 2.5. The noise variance is chosen as $\sigma^2 \approx -115$ dBm, which is thermal noise in addition to a 5 dB noise

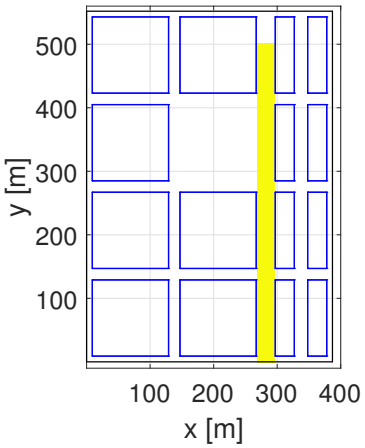

Figure 3.1: *An illustration of the Madrid grid, where the Gran-Via is highlighted in yellow.*

figure. The bandwidth is 240 kHz.

3.3 User Equipment

Throughout the simulations, the UE will be modeled as a vertical dipole. In the simulations, the UE will use transmit powers of 21 dBm and 10 dBm. These values follow from that, in LTE, 21 dBm is the maximum allowed transmit power, and 10 dBm was considered reasonable from a UE power consumption perspective. The UE will, throughout this thesis always be 1.5 m above the ground. This is assumed to be known at the base station. The orientation of the UE is fixed throughout the simulation, which is reasonable, since it is assumed to be mounted on a vehicle.

3.4 Transmission of Pilot Reference Signals

Tracking will be done using two different rates of transmitted pilot signals, 10 pilots per second, and 100 pilots per second. In massive MIMO, a pilot signal is sent in each coherence interval. The coherence time depends on the UE velocity and the propagation channel, but is in the order of a milli-second. In the simulator, there is a pilot signal opportunity every 0.2 ms. This means that with the pilot rates we have chosen, we utilize every fiftieth or five-hundredth pilot signal opportunity in our tracking.

3.5 Channel Model

The simulator uses a ray-tracing based model, meaning that we have a multipath channel. In a multipath channel, the transmitted signal reaches the receiver by several paths. An

illustration of a multipath channel can be seen in Figure 3.2. The ray-tracing takes the 3D-structure of the Madrid Grid into account [13].

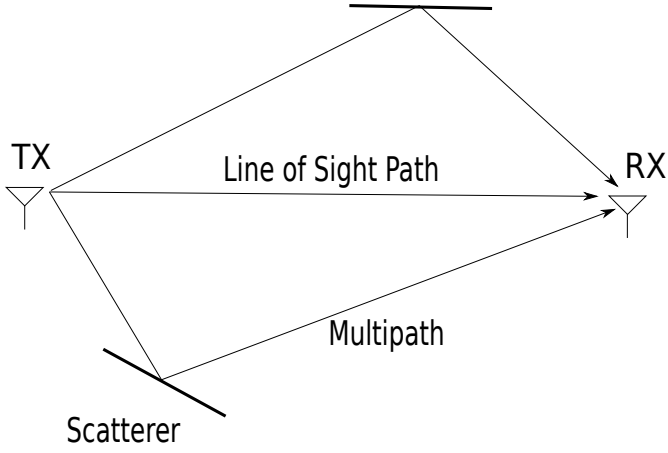

Figure 3.2: *An illustration of a multipath channel.*

Intuitively, one would assume that the channel gain would be smaller for positions further away on the highway. However, since multipath components are included in the simulations, this is not the case for us, which can be seen in figure 3.3. Note that the path loss

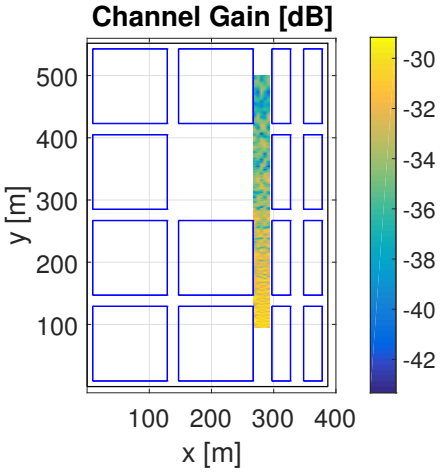

Figure 3.3: *Heat map of the channel gain for different positions on the Gran-Via, illustrating the effects of the multipath channel*

is smaller than in a typical outdoor scenario because the Gran-Via constitutes an urban canyon.

Since the simulator uses vertical dipoles both at the base station and at the UE, the signal

power received at the base station is extremely small when the UE is close to the base station or directly underneath it. This is why the bottom 20 % of the Gran-Via are not included in the presented results.

4

Simulations

In this chapter, the simulations that were done as part of this thesis work are presented and discussed. First we present simulations that were conducted in order to verify that the derived DML estimator attains the CRB in high SNR. Secondly, we give the simulations done in the simulator provided by Huawei Technologies, that is based on the METIS Madrid Grid.

4.1 Performance of the Derived DML Estimator

In this section we present a performance analysis of the DML estimator derived in section 2.2.2. The data used was generated using the exact model given in 2.5, for a two different antenna geometries (UCA and URA), using both 25 and 100 antennas for both geometries, where the 100 antenna array were used to represent a Massive MIMO antenna array. The carrier frequency used was 2.5 GHz. The same incident angles were used when generating the data for all antenna configurations considered. The MSEs were obtained by averaging the square estimation errors over 1000 realizations.

4.1.1 Uniform Cylindrical Array

In this section we compare the theoretical performance between two antenna arrays with a different number of antennas, but the same geometric structure. The Uniform Cylindrical Array (UCA) consists of several Uniform Circular Arrays, stacked on top of each other. Both the 25 and the 100 antenna array were comprised of five Uniform Circular Arrays, where each layer was composed of 5 and 20 antennas respectively.

We now investigate the performance of the derived DML estimator. We also investigate if there is an increase in performance by using an increased number of antennas. To do so, we compare the MSE and the CRB for variable SNR. We do this both for co-elevation

estimation and azimuth estimation, and for the 25 and 100 antenna arrays. This is depicted for co-elevation estimation and azimuth estimation in Figure 4.1 and Figure 4.2 respectively.

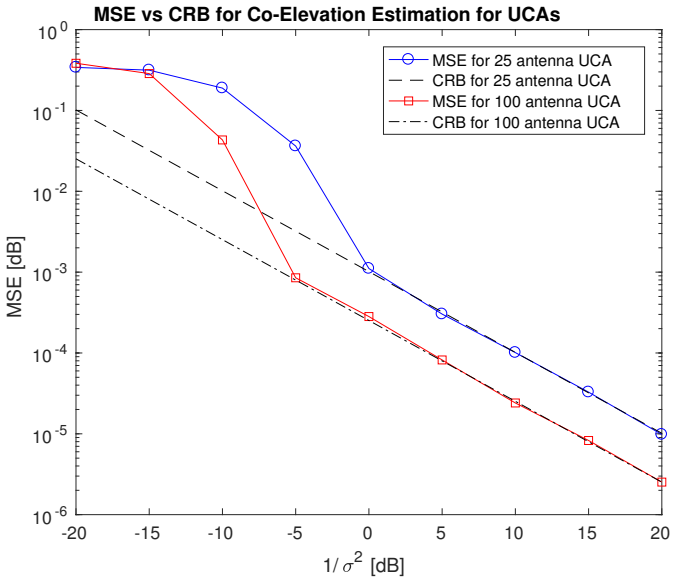

Figure 4.1: MSE*s vs* CRB*s in co-elevation estimation for variable* SNR *for* UCA *of* 25 *and* 100 *antennas respectively*

In these figures we clearly see that the CRBs are attained, in high SNR, both for co-elevation estimation and azimuth estimation, and also using both 25 antennas and 100 antennas. We also observe that the CRB is lower for the 100 antenna array than for the 25 antenna array, meaning that performance increases with the number of antennas.

4.1.2 Uniform Rectangular Array

In this section we compare the theoretical performance between two antenna arrays with a different number of antennas, but the same geometric structure. The Uniform Rectangular Array (URA) consists of $n \times n$ equispaced antennas in the same plane, i.e. the 25 and 100 URAs used are arranged in grids of 5×5 and 10×10 antennas respectively.

We now investigate the performance of the derived DML estimator, as well as investigate the increase in performance by using an increased number of antennas. To do so, we compare the MSE and CRB for variable SNR, both for co-elevation estimation and azimuth estimation, and for both using 25 and 100 antennas. This is depicted for co-elevation estimation and azimuth estimation in Figure 4.3 and Figure 4.4 respectively.

Just like for the UCA, we clearly see that the CRBs are attained, in high SNR, both for co-elevation estimation and azimuth estimation, and also using both 25 antennas and 100

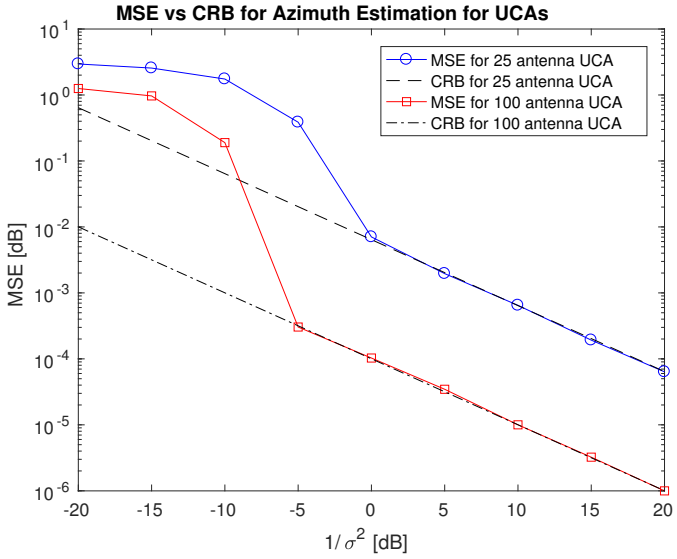

Figure 4.2: MSEs *vs* CRBs *in azimuth estimation for variable* SNR *for* UCA *of* 25 *and* 100 *antennas respectively*

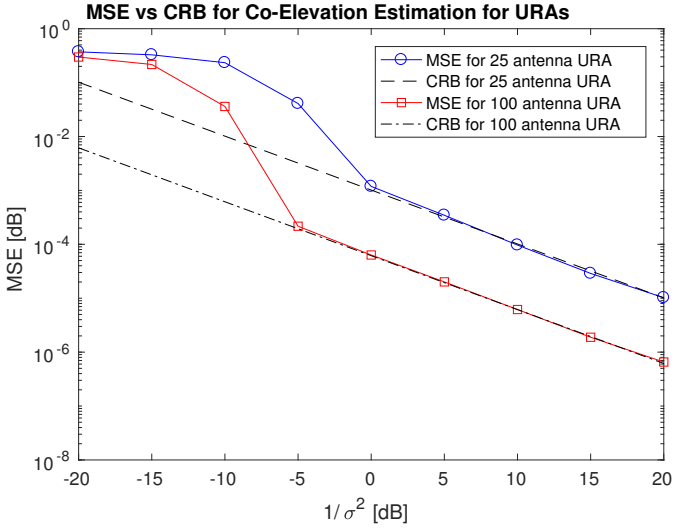

Figure 4.3: MSEs *vs* CRBs *in co-elevation estimation for variable SNR for Uniform Rectangular Arrays of* 25 *and* 100 *antennas respectively*

antennas. We also observe that the CRB is lower for the 100 antenna array than for the 25 antenna array, meaning that performance increases with the number of antennas.

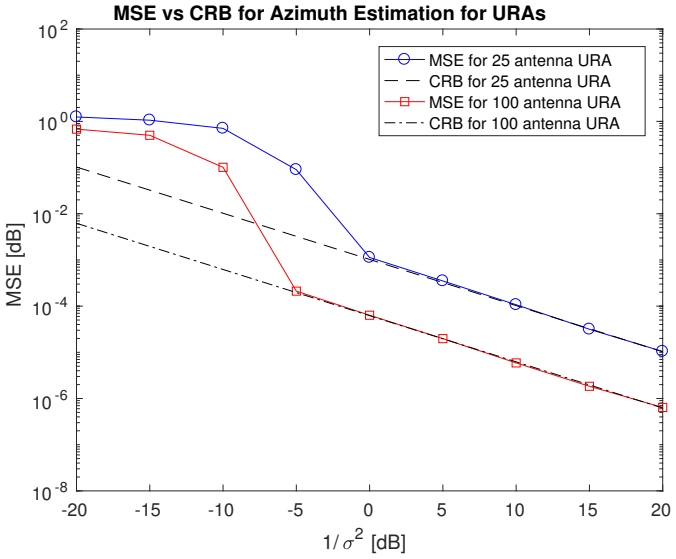

Figure 4.4: MSEs *vs* CRBs *in azimuth estimation for variable SNR for Uniform Rectangular Arrays of* 25 *and* 100 *antennas respectively*

4.1.3 Discussion

We have seen that the CRB is attained for the derived DML estimator for both antenna geometries, and for both numbers of antennas, that were studied.

For both antenna geometries, we also note that the CRB is lower for the antenna with more antennas. We conclude that an antenna array with more antennas is capable of more accurate positioning.

4.2 DOA Estimation, Position Estimation and Tracking on the Madrid Grid

In this section we present the simulation results from the simulator provided by Huawei Technologies. The simulation results are mainly presented using complementary cumulative distribution functions CCDFs. In order to visualize the RMSE for estimation of the different parameters for different positions on the Gran-Via, color plots (which we will call heat maps) are used. An example of a heat map can be found in Figure 3.3. To make small differences distinguishable, the errors equal to or larger than some threshold will be visualized as the same color (yellow). This truncation does disregard some information. However, this data is instead preserved in the CCDFs.

Since the base station will receive different signal powers for different distances to the UE, it is natural to present simulation results for different parts of the Gran-Via. We will divide the Gran-Via into three parts. These are called *Cell A*, *Cell B* and *Cell C*. Cell A

is the full Gran-Via, Cell B is the upper half of the Gran-Via (300 m-500 m from the base station) and Cell C is the lower half of the Gran-Via (100 m-300 m from the base station). A visualization of the three cells can be seen in Figure 4.5

Figure 4.5: *Cell A, Cell B and Cell C highlighted on the Gran-Via.*

In all simulations, the car starts at the far end of the Gran-Via and drives towards the base station in a straight line at 50 km/h.

Instead of presenting plots for all possible parameter configurations, e.g. 25 and 100 antennas, 10 dBm and 21 dBm UE transmit power, we will reduce the number of plots presented by doing head to head comparisons. For instance, we will compare the positioning performance for the 25 antenna array to the positioning performance of the 100 antenna array. After this comparison is made, we will only use the better one, since we are only interested in the best possible positioning performance we can achieve. Not all plots will be presented in these comparisons, but for all parameter configurations that are discussed, all CCDF plots and all heat maps can be found in the appendix.

4.2.1 Performance with Different Number of Antennas

In this section we compare positioning performance between base stations of 25 and 100 antennas, in order to investigate whether we obtain a performance gain by increasing the number of antennas.

The comparison is made by studying the CCDF of the position estimation error for both number antennas for a fixed transmit power of 10 dBm, and the same method for position estimation (DML). These CCDFs for 100 and 25 antenna arrays can be found in Figure 4.6 and Figure 4.7 respectively.

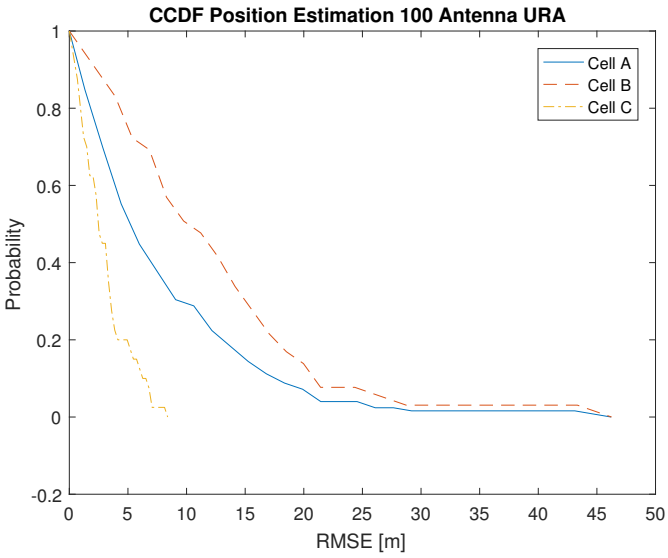

Figure 4.6: CCDF *for a 100 antenna array using the* DML *and a transmit power of* 10 *dBm.*

By studying these figures we conclude that the antenna array with 100 antennas yields better performance for positioning, e.g. the 100 antenna array can provide positioning with an error less than 15 m in 80 % of all cases, while the 25 antenna array can only

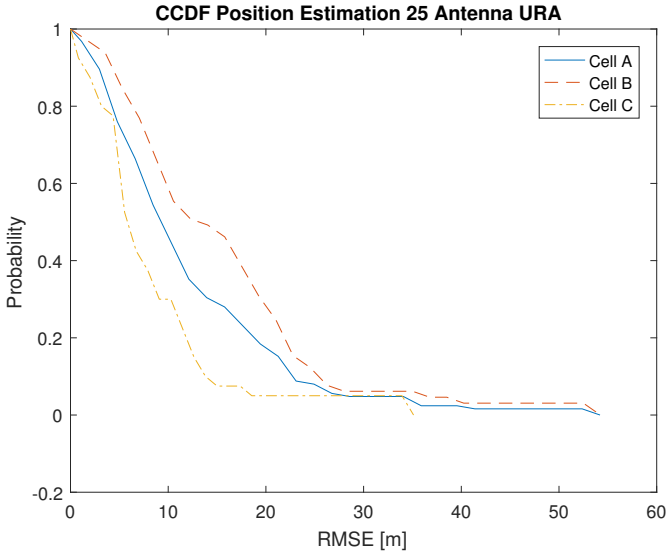

Figure 4.7: CCDF *for a 25 antenna array using the* DML *and a transmit power of* 10 *dBm.*

provide an error less than 20m in 80 % of all cases.

Since we have concluded that the 100 antenna array provides better performance, we will only use the 100 antenna array in the coming simulations, and instead vary other parameters, such as the transmit power.

4.2.2 Performance with Different Transmit Powers

In this section we compare positioning performance for two different UE transmit powers, 10 dBm and 21 dBm. We do this in order to investigate whether we could obtain a performance gain by an increase in UE transmit power.

The comparison is made by studying the CCDF of the position estimation error for both transmit powers for a 100 antenna array, and the same method for position estimation (DML). The CCDF:s for 10 dBm and 21 dBm UE transmit powers can be found in Figure 4.6 and Figure 4.8.

By studying these two figures, we can conclude that there is no significant gain from using a UE transmit power of 21 dBm over a UE transmit power of 10 dBm. The performance using these two transmit powers is almost identical, while the UE has roughly ten times the transmit power. Since energy consumption is a concern of the UE we will henceforth, without significant loss in positioning accuracy, only use a UE transmit power of 10 dBm.

One reason for not observing a performance improvement with increasing signal power is believed to be that we are operating in the high-SNR regime, where the errors in positioning are caused by errors in our model instead of by the additive noise. Our estimator

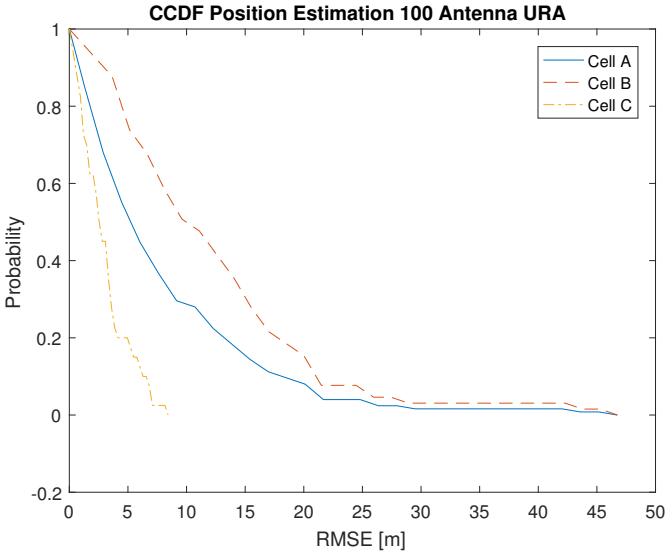

Figure 4.8: CCDF *for a 100 antenna array using the* DML *and a transmit power of* 21 *dBm.*

assumes that all antennas in the array see the same DOA, while this is not the case in reality, nor in the channel generation.

4.2.3 Performance of Filtering with Different Update Intervals

In this section, we compare the positioning performance using EKFs with different update intervals, both using the constant angular acceleration model. In every transmission time interval (TTI), there is a pilot signal opportunity. One TTI corresponds to the coherence time. The different update intervals are given by how many TTIs there are in between two utilized pilot signals. The two update intervals used are 50 TTI and 500 TTI, corresponding to utilizing 100 pilot signals and 10 pilot signals, respectively, for positioning every second, assuming a 0.2 ms coherence time. The comparison will be made for a 100 antenna array and for a 10 dBm UE transmit power.

The positioning error CCDFs for EKFs with update intervals of 50 TTI and 500 TTI can be found in Figure 4.9 and Figure 4.10 respectively. By studying these two figures we can conclude that the EKF using an update interval of 500 TTI yields a slightly better result for Cell C, where it could guarantee a positioning error less than 8 m. However, the EKF using the 50 TTI update interval has a CCDF with a slightly less heavy tail, with a worst case positioning error of 47.5 m. The EKF using the 50 TTI update interval is expected to perform better, since it would utilize more measurements, an effect expected to be more apparent for higher velocities of the UE. The reason why it is not achieving better accuracy for the entire highway in our simulations, compared to the 500 TTI EKF, is believed to be due to model errors and tuning of the state noise. Since the performance of the two filters was so similar, it is difficult to deduce which one is better. However, in

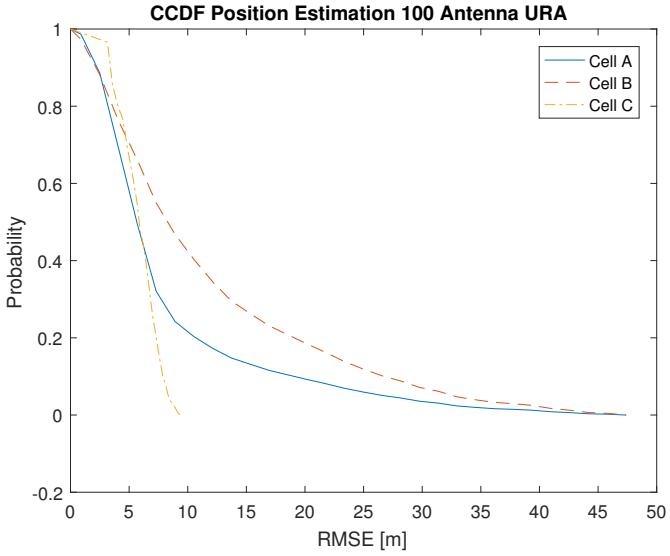

Figure 4.9: CCDF *for a 100 antenna array using the* EKF *with* 50 TTI *update interval and a transmit power of* 10 *dBm.*

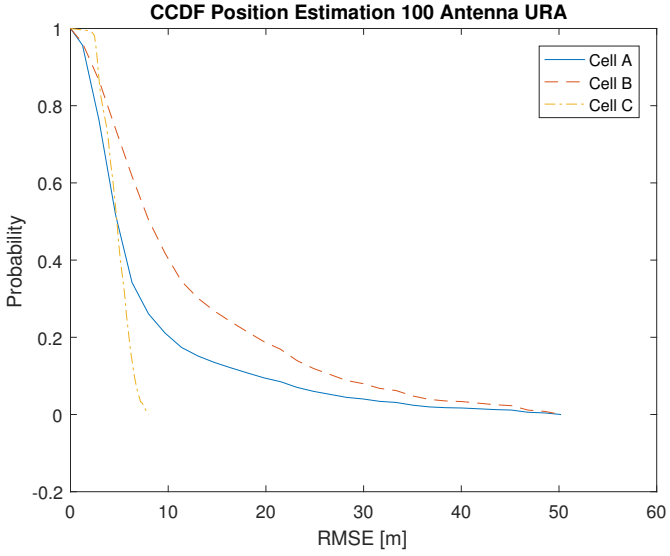

Figure 4.10: CCDF *for a 100 antenna array using the* EKF *with* 500 TTI *update interval and a transmit power of* 10 *dBm.*

future comparisons, we chose to only use the 500 TTI EKF, due to its accuracy in Cell C, which is the highest we have achieved so far.

4.2.4 Performance of Filtering and Batch Estimation

In this section we compare positioning performance using an EKF with 500 TTI update interval, and positioning performance using only a DML estimate. We do this in order to investigate whether we could obtain a performance gain by using an EKF over only using DML estimates. The EKF is expected to perform better, since it accounts for how the vehicle is expected to move, and utilizes previous state estimates.

We start by studying the CCDF of the position estimation error both using the EKF and using the DML estimator. These can be found in Figure 4.10 and Figure 4.6. Even though there is not a large difference, the EKF outperforms the DML in Cell C, e.g. the EKF can with absolute certainty perform positioning with an error less than $\approx$ 8 m, while the DML guarantes an error less than ≈ 8.5 m. Remember that the EKF was expected to outperform the DML. For Cell A however, the DML could perform positioning with an error of less than 18.5 m with 90 % certainty, while the EKF could only achieve an error less than ≈ 19.5 m with the same certainty.

In order to explain why the DML is sometimes better than the EKF, we study the CCDF of the co-elevation estimation error, both using the DML and the EKF. These CCDFs are depicted in Figure 4.11 and Figure 4.12, respectively. In these plots, we observe two

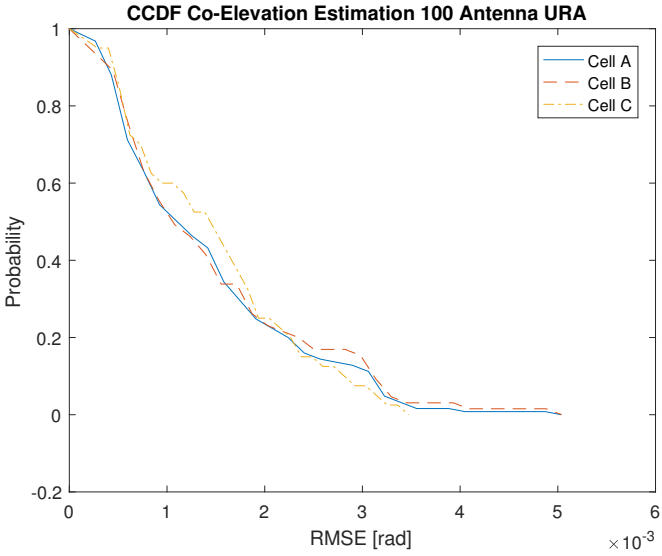

Figure 4.11: CCDF *for a 100 antenna array using* DML *and a transmit power of* 10 *dBm.*

things that one might consider strange. The DML seems to yield similar accuracy both in Cell B and Cell C, even though the base station receives less signal power in Cell B. This can be explained by the fact that the model errors outweigh the estimation errors (i.e. that we are operating in a high-SNR regime). Secondly, and more interestingly, from Figure 4.9 we get that the positioning using the EKF was more accurate in Cell C than in Cell B, but

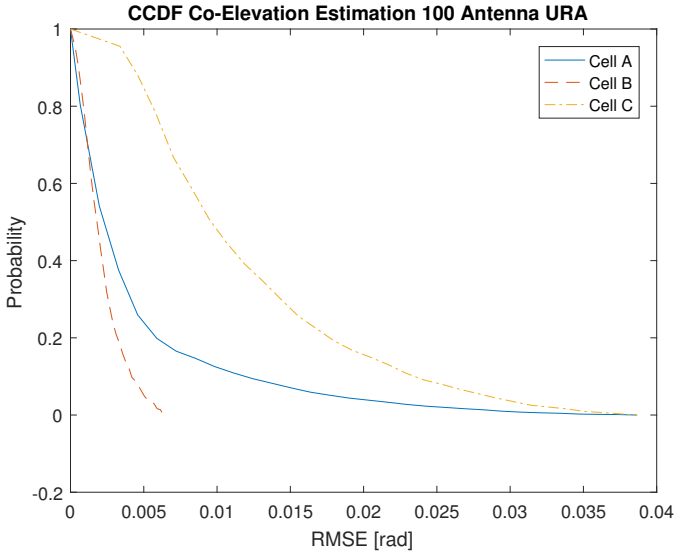

Figure 4.12: CCDF *for a 100 antenna array using the* EKF *with* 500 TTI *update interval and a transmit power of* 10 *dBm.*

from Figure 4.12 we see that estimation of the co-elevation DOA is more accurate in Cell B. Moreover, if we instead study the CCDFs of the azimuth estimation errors for both DML and the EKF, depicted in Figure 4.13 and Figure 4.14, we see that the DML again yields similar performance in both cells, while the EKF has more accurate azimuth DOA estimation in Cell C. This is because the filter was tuned based on positioning accuracy for the full highway, where accuracy in co-elevation estimation is more important further away and accuracy in azimuth estimation has more impact nearby.

We will now present an illustration regarding tuning the state noise. Consider the simplified scenario depicted in Figure 4.15. Here, the UE is moving towards the base station with velocity $v = 50$ km/h ≈ 14 m/s. The height of the UE and base station is 1.5 m and 50 m respectively. The UE is assumed to be at distance d meters from the base station at time $t = 0$. The DOA, θ can then be written as

$$\theta(t, d) = \arctan\left(\frac{48.5}{d - 14t}\right). \tag{4.1}$$

The design parameter when using the constant angular acceleration model with the EKF is the state noise, which corresponds to the third derivative with respect to time of θ, $\dddot{\theta}$. Estimates of the appropriate process noise magnitude for the different parts of the highway can be obtained by evaluating $\dddot{\theta}(0, d)$ for $d = 100$ m and $d = 500$ m respectively. We obtain $\dddot{\theta}(0, 100) \approx 4 \cdot 10^{-3}$ rad/s^3 and $\dddot{\theta}(0, 500) \approx 1.2 \cdot 10^{-5}$ rad/s^3. This means that the state noise should have different orders of magnitude for different parts of the highway, at least when tracking the co-elevation.

The state noise was tuned to obtain as good position estimation CCDF for the entire Gran-

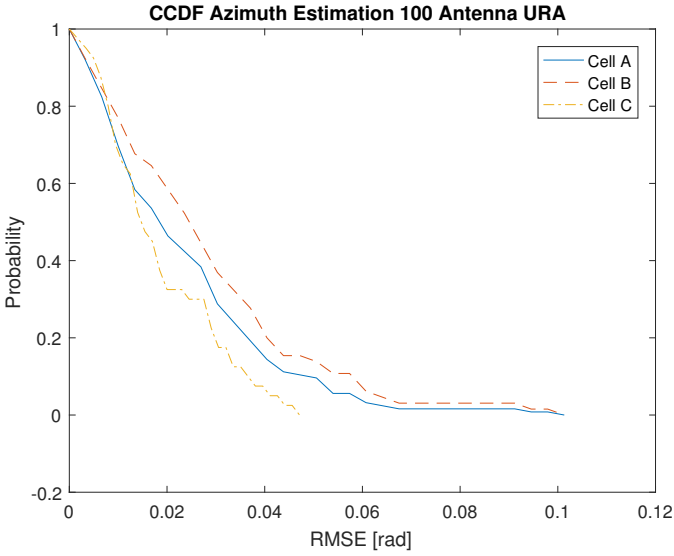

Figure 4.13: CCDF *for a 100 antenna array using* DML *and a transmit power of* 10 *dBm.*

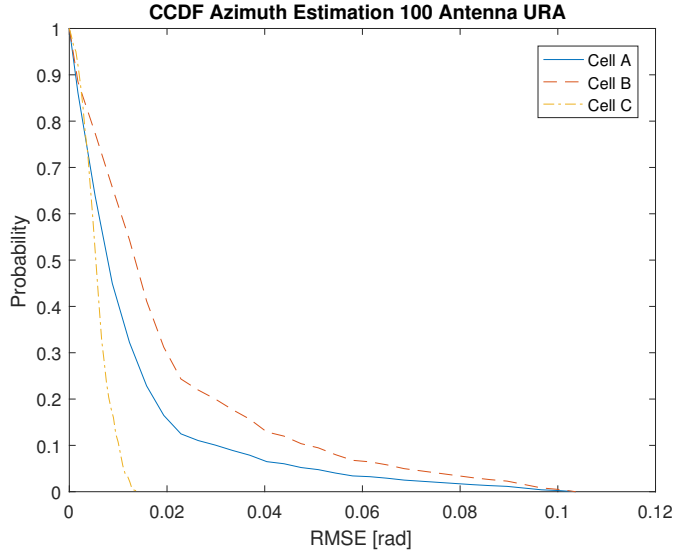

Figure 4.14: CCDF *for a 100 antenna array using the* EKF *with* 500 TTI *update interval and a transmit power of* 10 *dBm.*

Via, but since the state noise for the change in co-elevation should be of different orders of magnitude for different parts of the Gran-Via, the state noise for the tuned filter is

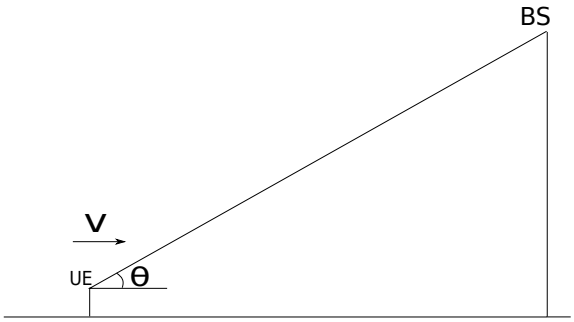

Figure 4.15: *A simplified overview of the simulation setup.*

a compromise between a filter that is good for the parts near the base station and the parts further away. Another issue is that the state noise for both co-elevation and azimuth is identical in our model, which is not realistic, since, in our case, the co-elevation is expected to change, while the azimuth is expected to remain somewhat constant. We have now given a reason as to why the EKF is not outperforming the DML as would be expected.

When tuning the state noise, it is much more important for the EKF to provide accurate co-elevation estimation for the parts of the highway that are far away from the base station, since there a small co-elevation DOA estimation error gives rise to a larger positioning error than near the base station. This is depicted in Figure 4.16. Using the notation in

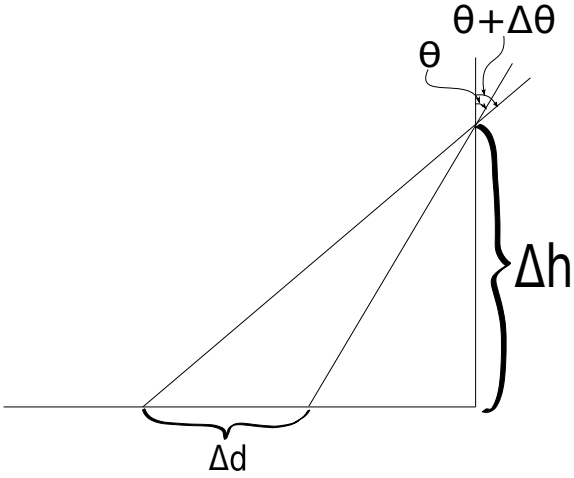

Figure 4.16: *An illustration of the positioning error caused by an error in estimation of the co-elevation* DOA

Figure 4.16, we obtain an expression for the positioning error Δd caused by an estimation error of the co-elevation DOA, $\Delta\theta$ as

$$\Delta d = \Delta h(\tan(\theta + \Delta\theta) - \tan(\theta)). \tag{4.2}$$

A plot of Δd for $0 \leq \theta < \frac{\pi}{2}$, with $\Delta\theta = 0.01$ rad, can be seen in Figure 4.17. We

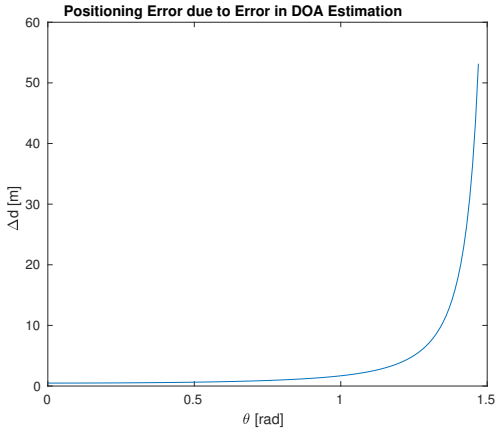

Figure 4.17: *An plot of the positioning error caused by an error in estimation of the co-elevation* DOA.

conclude that since it is of much greater importance that the state noise is tuned so that the EKF performs well, co-elevation-wise, in the parts of the highway that are far away, the best EKF, which was tuned with respect to positioning accuracy for the full highway, has better co-elevation estimation performance in these parts.

The positioning error due to error in azimuth estimation is proportional to the distance between the UE and the base station. For the areas that are far away from the base station, this error is dominated by the error caused by errors in the co-elevation estimation. This means that the errors in azimuth estimation are more significant in areas near the base station, where the errors in co-elevation estimation have less impact. This is why the tuned filter has better accuracy in azimuth estimation in the parts of the highway that is near the base station.

In order to improve performance, we could e.g. either estimate the state noise (by having it as a state), or a filter bank, where many separate Kalman filters are used, and then choosing the one with the smallest prediction error, see e.g [9].

The results from the simulations presented in this section also makes it possible to draw conclusions regarding the deployment of base stations. We showed that, using a massive MIMO base station of 100 antennas we could achieve a positioning error of less than 8 m when the UE was 100 m-300 m from the base station. Thus, if this accuracy is considered satisfactory, one should deploy base stations every 600 m. If the accuracy achieved for the full highway, where we could guarantee a positioning error of less than 18.5 m with a 90% certainty, is considered satisfactory, then it would be enough to deploy base stations so that it is at most 1000 m in between them. Significant improvement of positioning accuracy is believed to be possible by using more sophisticated filtering methods.

5
Conclusion

In this chapter we conclude the thesis. We present the conclusions we have made, but also give some suggestions for future work, that should improve positioning performance even more.

5.1 Concluding Remarks

In this thesis the performance of massive MIMO systems for positioning, using both filtering and batch estimation was studied. By simulations, we have found answers to the questions we aimed to answer in this thesis:

1. *What is the positioning accuracy a massive* MIMO *system can achieve in open highways*

   Using the EKF we managed to achieve a position estimation error of less than 8 meters when the UE was near (100 m-300 m) the base station. This is a result that most likely could be improved by using better filtering methods, where the state noise is varying, or where different models could be used for different parts of the highway.

   We also saw that using a UE transmit power of 21 dBm instead of a UE transmit power of 10 dBm yields no significant improvement in positioning performance. This means that we can reduce power consumption in the UE, without sacrificing positioning accuracy.

2. *Does increasing the number of base station antennas improve positioning accuracy?*

   We concluded that positioning accuracy was improved, both in theory and practice,

when using a larger number of antennas in the base station array.

3. *How should base stations be deployed to maintain a high positioning accuracy?*

   The base station deployment depends on the desired positioning accuracy. If an accuracy of 8 m is needed, then base stations should be deployed every 600 m. If instead it would be considered acceptable to have a positioning error less than 18.5 m with a 90 % certainty, then it is sufficient for the base stations to be deployed every 1000 m.

5.2 Future Work

In the very beginning of this thesis, a number of limitations were posed on its scope. It is therefore natural to let these limitations be the starting point for future work. A list of suggestions for future work follows. In particular,

- only positioning and tracking of a single vehicle is studied. In order to ensure that the results in this thesis are also achievable in practice, one should study the case where there are pilot contamination from UEs in neighbouring cells.
- positioning was done by assuming the height of the UE, and we saw that a small error in the estimated DOA could result in a large positioning error. It would therefore be interesting to study what positioning performance we could obtain, if we could take the curvature of the wavefront into account.
- we concluded that using the same process noise for the entire highway did not yield better results than the ones obtained from raw DML. It is therefore natural to study what performance we could achieve if we were to use a better filtering method, such as an adaptive Kalman filter or a filter bank approach.
- we only studied the performance gain of using more antennas in the antenna array. It would also be interesting to study the positioning performance for different antenna geometries and for different apertures.
- parameterization of the steering vector was only done by the co-elevation and azimuth DOAs. It would be interesting to study the positioning performance if we instead were to parameterize the steering vector by the position of the vehicle, it is possible that we could circumvent the large positioning errors caused by small errors in the co-elevation estimation for positions far away on the Gran-Via.

Appendix

A

Simulation Plots

In this chapter of the appendix present some plots that were omitted in the thesis itself. We give CCDFs and heat maps for the position estimation error, the co-elevation estimation error, and the azimuth estimation error, for all parameter configurations that were used in the simulations chapter. The reason that some plots are given both here and in the thesis is to aid in interpretation of the simulation results. The plots for the different configurations are given in the same order as they appear in the thesis.

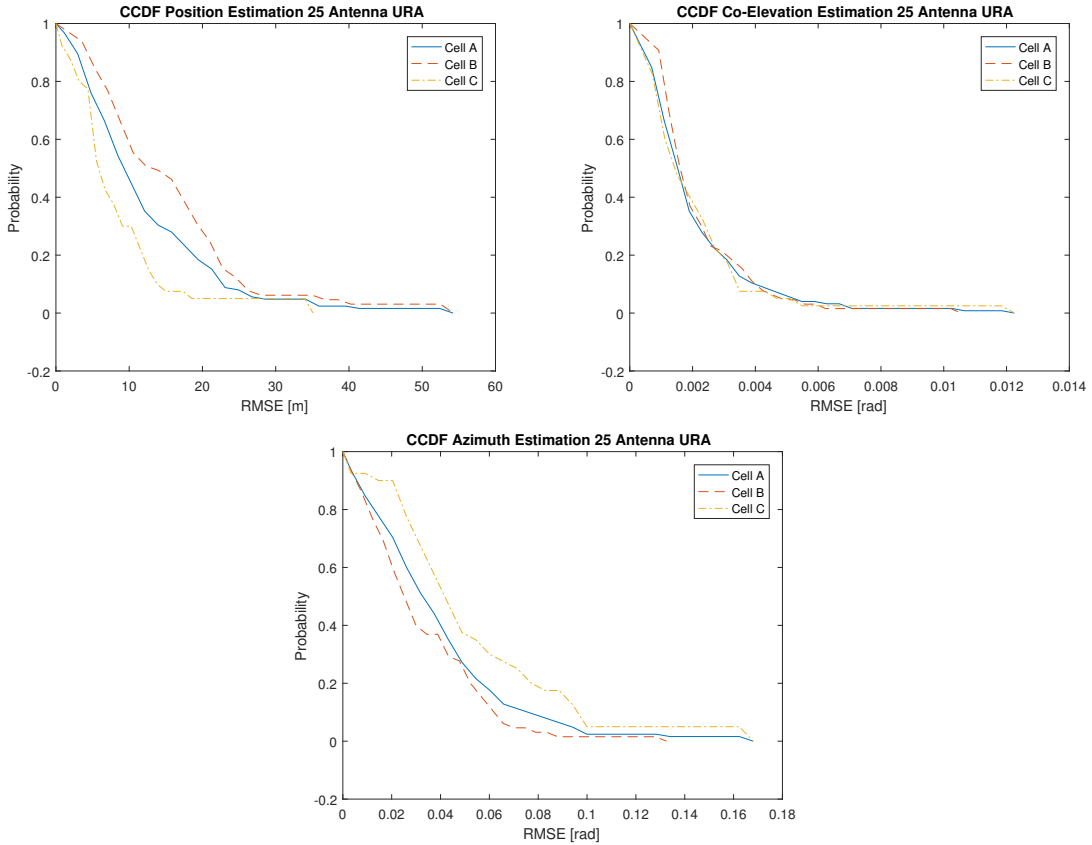

Figure A.1: CCDFs *for position, co-elevation and azimuth estimation for a 25 antenna array using the* DML *and a transmit power of* 10 *dBm.*

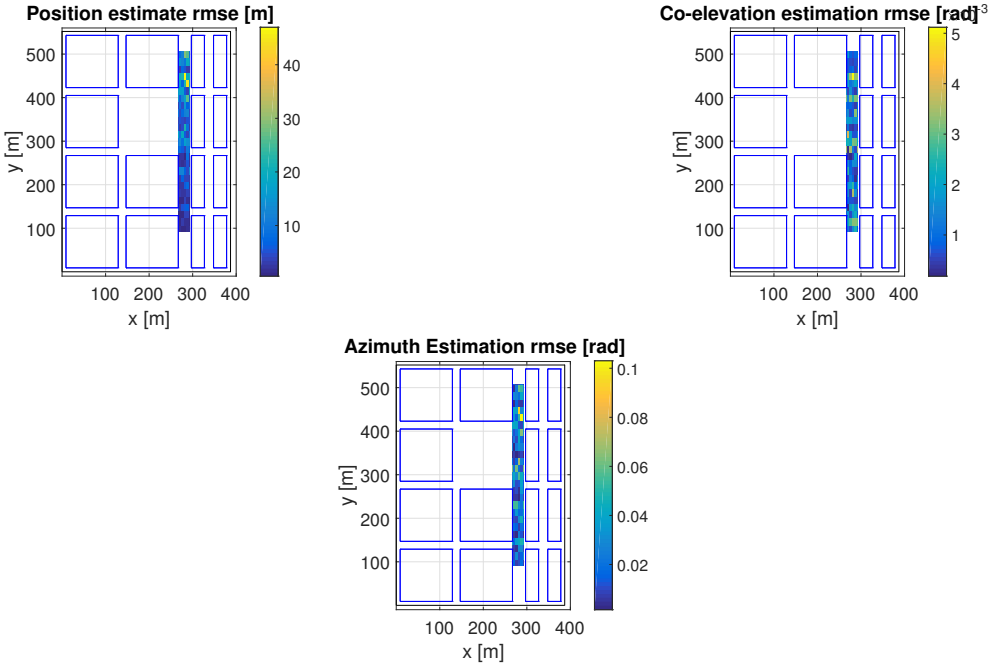

Figure A.2: *Heat maps for position, co-elevation and azimuth estimation for a 25 antenna array using the* DML *and a transmit power of* 10 *dBm.*

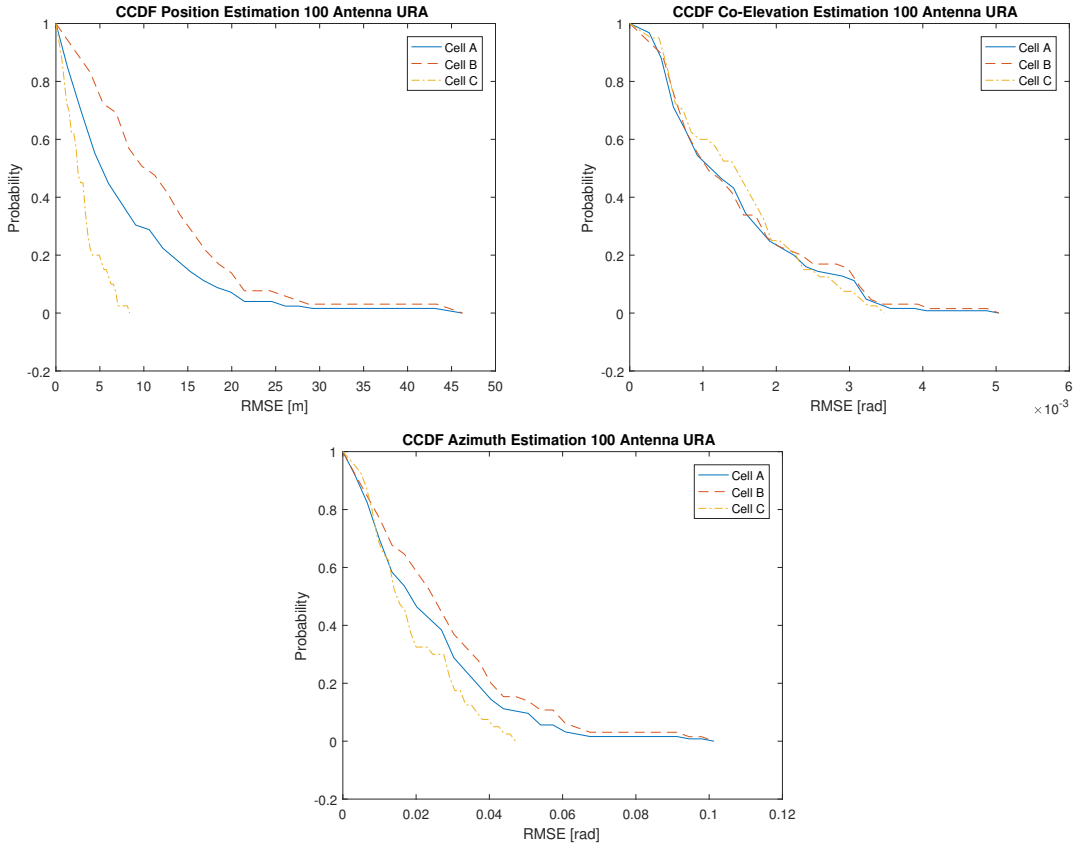

Figure A.3: CCDFs *for position, co-elevation and azimuth estimation for a 100 antenna array using the* DML *and a transmit power of* 10 *dBm.*

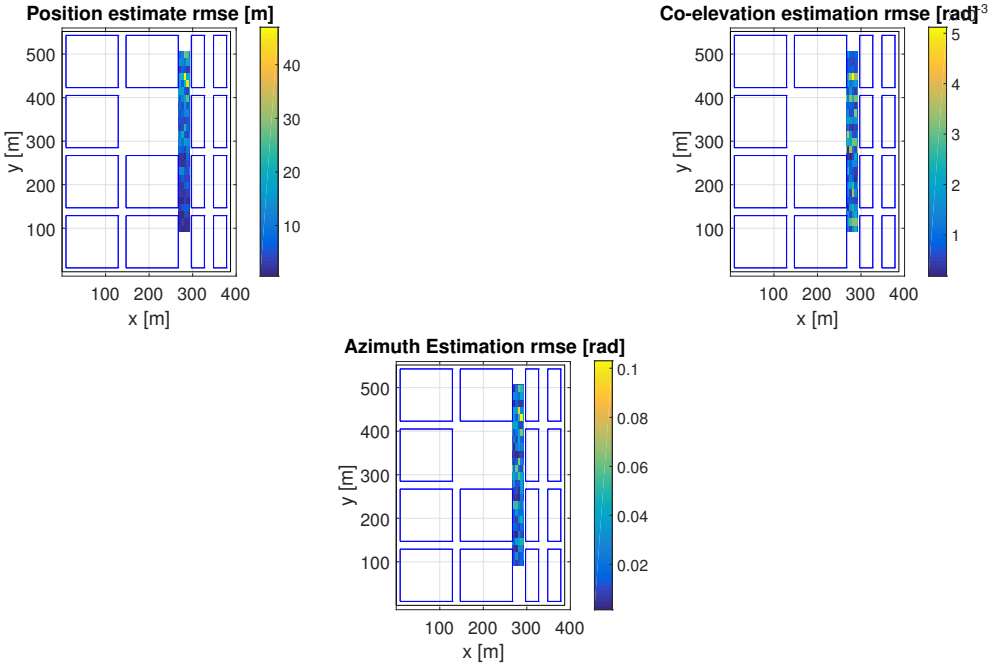

Figure A.4: *Heat maps for position, co-elevation and azimuth estimation for a 100 antenna array using the* DML *and a transmit power of* 10 *dBm.*

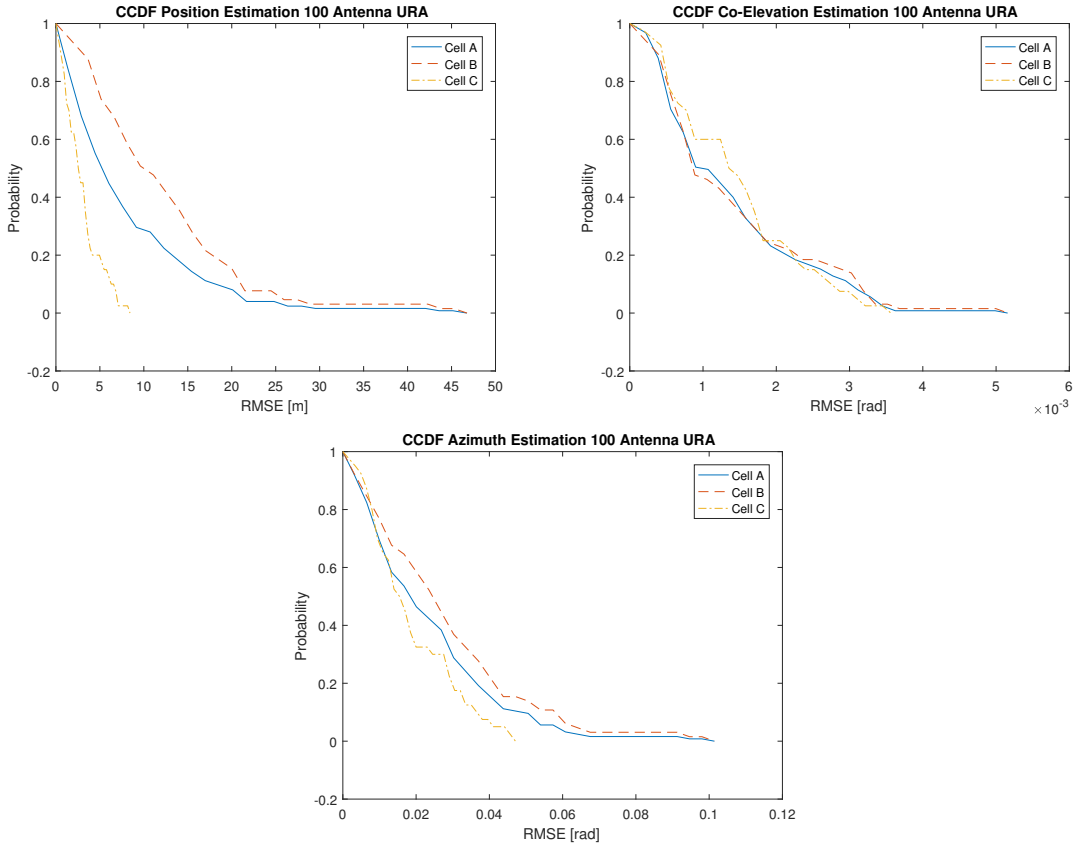

Figure A.5: CCDFs *for position, co-elevation and azimuth estimation for a 100 antenna array using the* DML *and a transmit power of* 21 *dBm.*

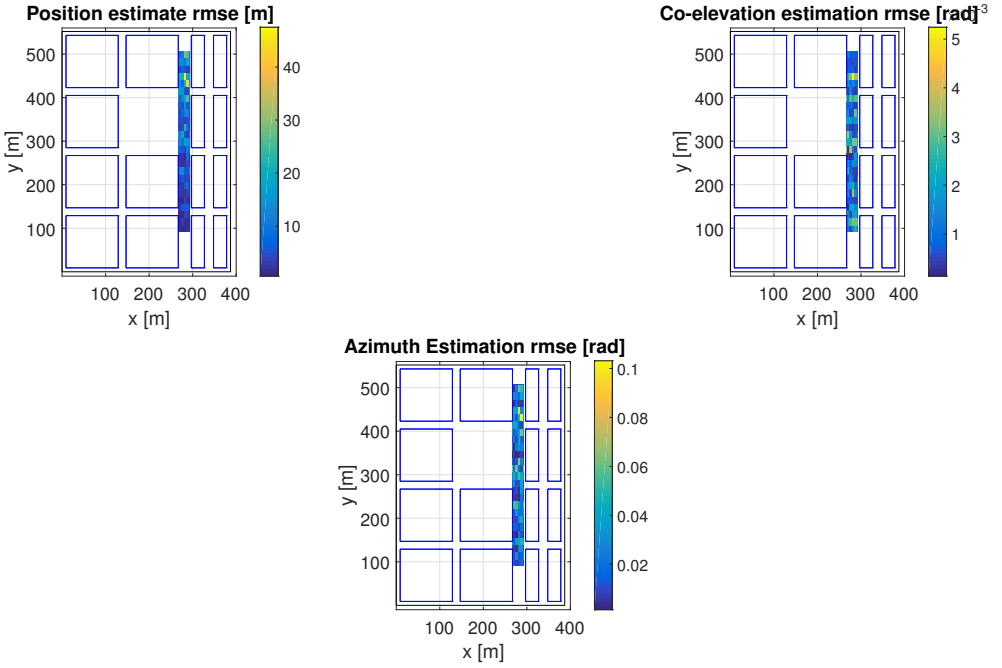

Figure A.6: *Heat maps for position, co-elevation and azimuth estimation for a 100 antenna array using the* DML *and a transmit power of* 21 *dBm.*

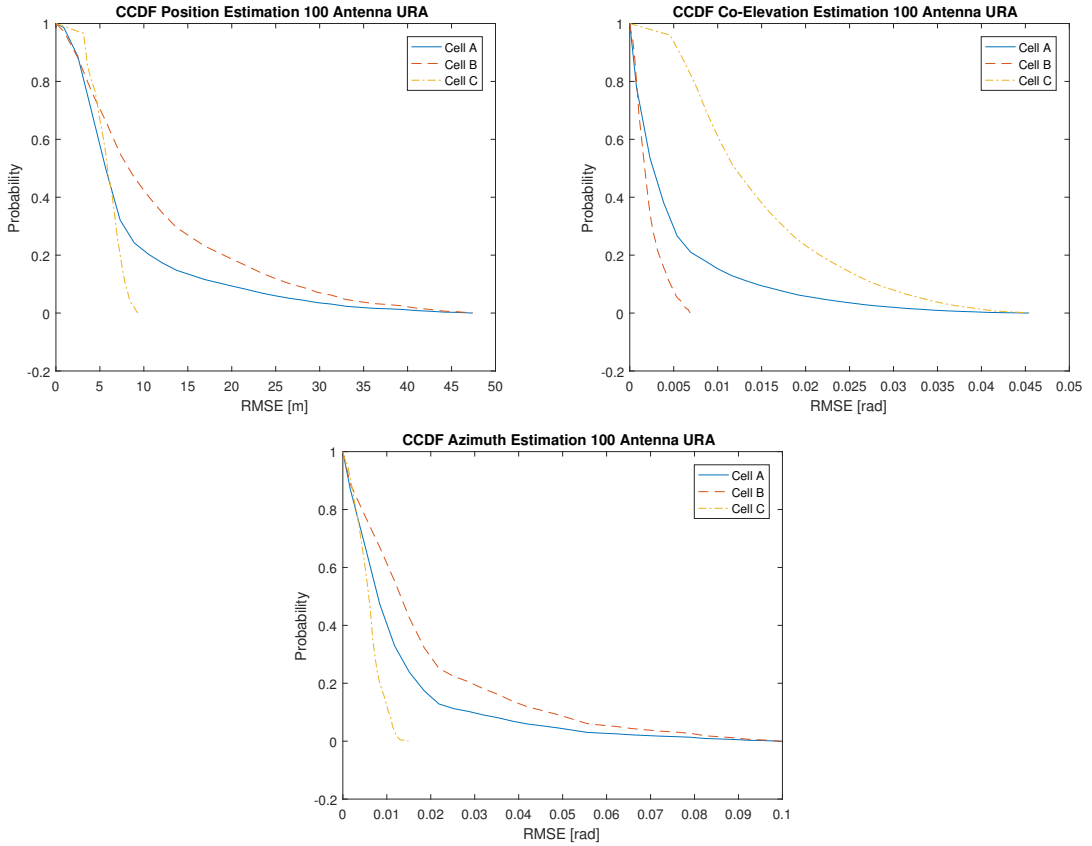

Figure A.7: CCDFs *for position, co-elevation and azimuth estimation for a 100 antenna array using the* EKF *with an update interval of 50 TTI and a transmit power of* 10 *dBm.*

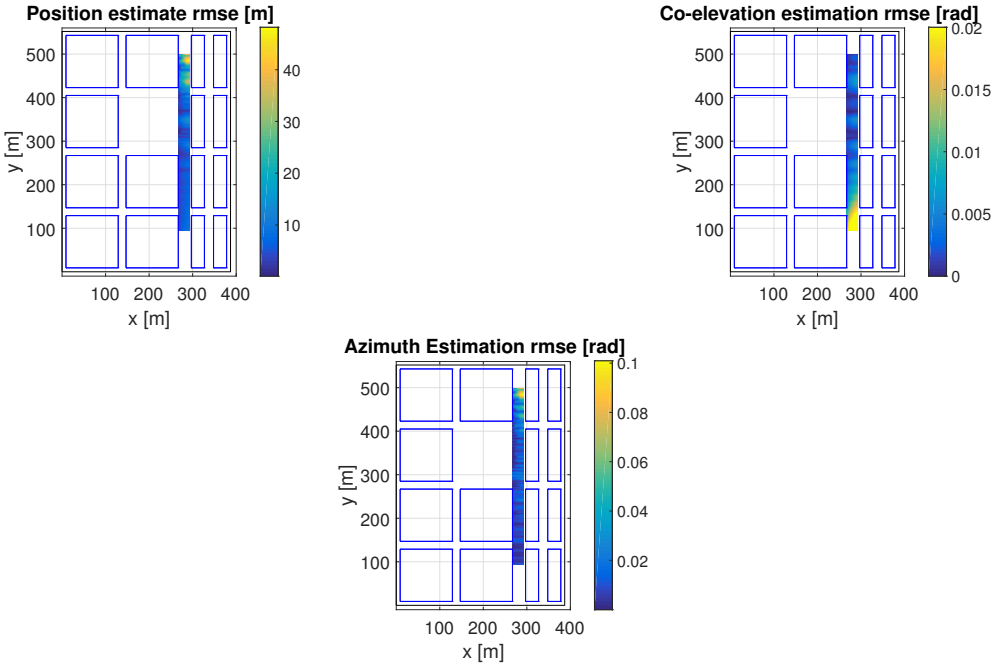

Figure A.8: *Heat maps for position, co-elevation and azimuth estimation for a 100 antenna array using the* EKF *with an update interval of 50 TTI and a transmit power of* 10 *dBm.*

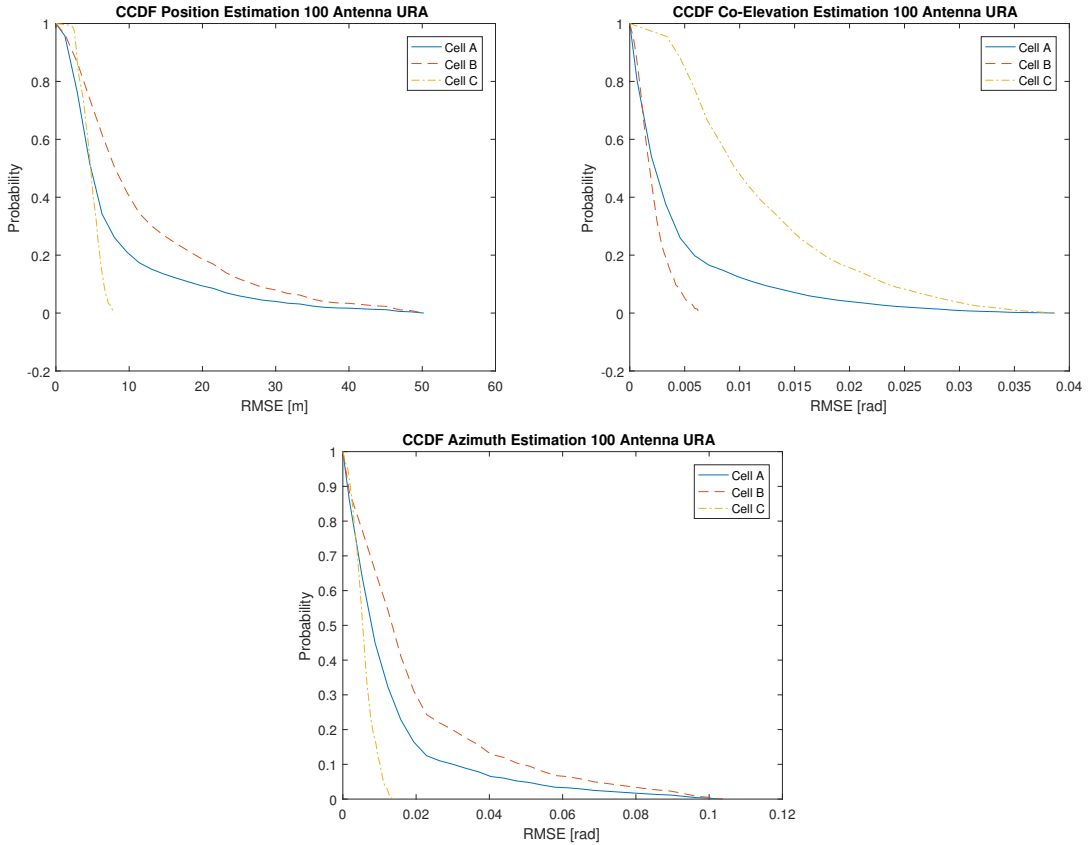

Figure A.9: CCDFs *for position, co-elevation and azimuth estimation for a 100 antenna array using the* EKF *with an update interval of 500 TTI and a transmit power of* 10 *dBm.*

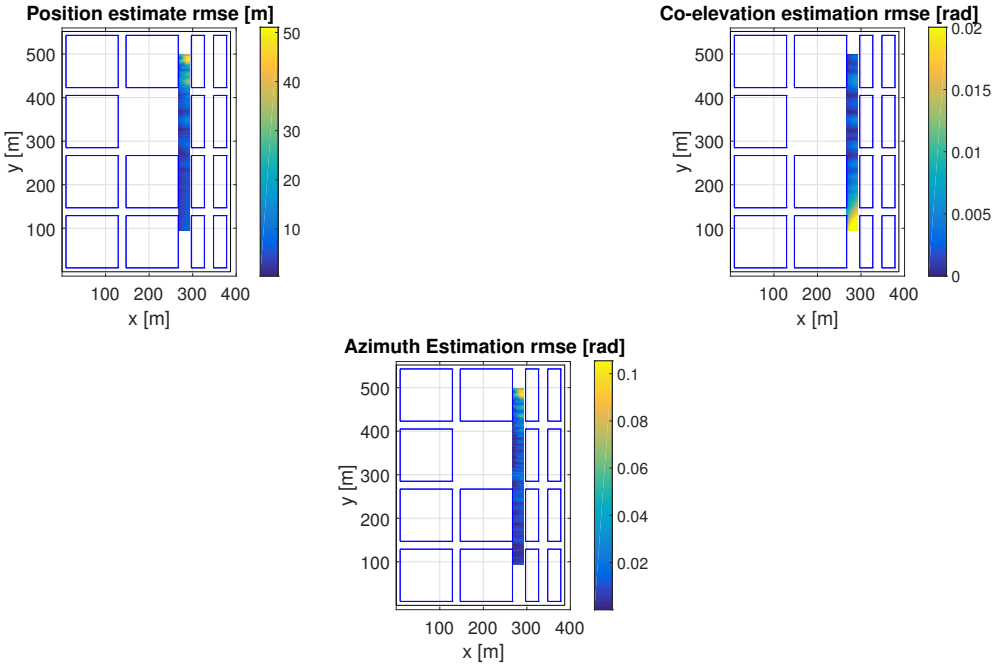

Figure A.10: *Heat maps for position, co-elevation and azimuth estimation for a 100 antenna array using the* EKF *with an update interval of 500 TTI and a transmit power of* 10 *dBm.*

Bibliography

[1] Fabio Belloni, Andreas Richter, and Visa Koivunen. Doa estimation via manifold separation for arbitrary array structures. *IEEE Transactions on Signal Processing*, 55(10):4800–4810, 2007. Cited on page 8.

[2] Emil Björnson, Erik G Larsson, and Thomas L Marzetta. Massive MIMO: Ten myths and one critical question. *IEEE Communications Magazine*, 54(2):114–123, 2016. Cited on page 1.

[3] Gunnar Blom, Jan Enger, Gunnar Englund, Jan Grandell, and Lars Holst. *Sannolikhetsteori och statistikteori med tillämpningar*. Studentlitteratur, 5 edition, 2004. Cited on page 11.

[4] Volvo Cars. Drive me – the self-driving car in action. URL `http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving/drive-me`. Accessed: 2017-05-18. Cited on page 2.

[5] Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021 white paper. URL `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html`. Cited on page 1.

[6] Mário Costa, Andreas Richter, and Visa Koivunen. Unified array manifold decomposition based on spherical harmonics and 2-d fourier basis. *IEEE Transactions on Signal Processing*, 58(9):4634–4645, 2010. Cited on pages 9 and 10.

[7] Mário Jorge Costa. Wavefield modeling and signal processing for sensor arrays of arbitrary geometry. 2013. Cited on page 9.

[8] Miriam A. Doron and Eyal Doron. Wavefield modeling and array processing, part i, part ii and part iii. *IEEE Trans. Signal Processing*, 42:2549–2580, 1994. Cited on pages 8 and 9.

[9] Fredrik Gustafsson. *Statistical sensor fusion*. Studentlitteratur, 2010. Cited on pages 14, 15, 17, 19, 20, 21, 22, and 43.

[10] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. Cited on page 20.

[11] Dagens Industri Karin Olander. Volvos och autolivs zenuity rullar. URL `http://www.di.se/bil/volvos-och-autolivs-zenuity-rullar/`. Accessed: 2017-05-18. Cited on page 2.

[12] H. Kim and Mats Viberg. Two decades of array signal processing research. *IEEE signal processing magazine*, 13(4):67–94, 1996. Cited on pages 2, 11, and 14.

[13] Mike Koivisto, Mário Costa, Janis Werner, Kari Heiska, Jukka Talvitie, Kari Leppänen, Visa Koivunen, and Mikko Valkama. Joint device positioning and clock synchronization in 5g ultra-dense networks. *arXiv preprint arXiv:1604.03322*, 2016. Cited on pages 21, 25, and 27.

[14] Jan Lundgren, Mikael Rönnqvist, and Peter Värbrand. Optimeringslära, studentlitteratur, 2003. Cited on pages 17 and 18.

[15] Thomas L Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Transactions on Wireless Communications*, 9(11):3590–3600, October 2010. Cited on page 1.

[16] METIS. D6.1 simulation guidelines, 2013. URL `https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D6.1_v1.pdf`. Accessed: 2017-05-16. Cited on page 25.

[17] NGMN. Nmgn 5g white paper, 2015. URL `https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf`. Cited on page 2.

[18] Andreas Richter. Estimation of radio channel parameters: Models and algorithms. ISLE, 2005. Cited on pages 14, 15, 18, and 19.

[19] Jagdish S Rustagi. *Optimization techniques in statistics*. Academic Press, Inc., 1994. Cited on page 19.

[20] Jussi Salmi, Andreas Richter, and Visa Koivunen. Detection and tracking of mimo propagation path parameters using state-space approach. *IEEE Transactions on Signal Processing*, 57(4):1538–1550, 2009. Cited on page 21.

[21] Mats Viberg, Björn Ottersten, and Thomas Kailath. Detection and estimation in sensor arrays using weighted subspace fitting. *IEEE transactions on Signal Processing*, 39(11):2436–2449, 1991. Cited on pages 5, 18, and 22.

[22] Joao Vieira, Steffen Malkowsky, Karl Nieman, Zachary Miers, Nikhil Kundargi, Liang Liu, Ian Wong, Viktor Öwall, Ove Edfors, and Fredrik Tufvesson. A flexible 100-antenna testbed for massive MIMO. In *Globecom Workshops (GC Wkshps), 2014*, pages 287–293. IEEE, 2014. Cited on page 1.